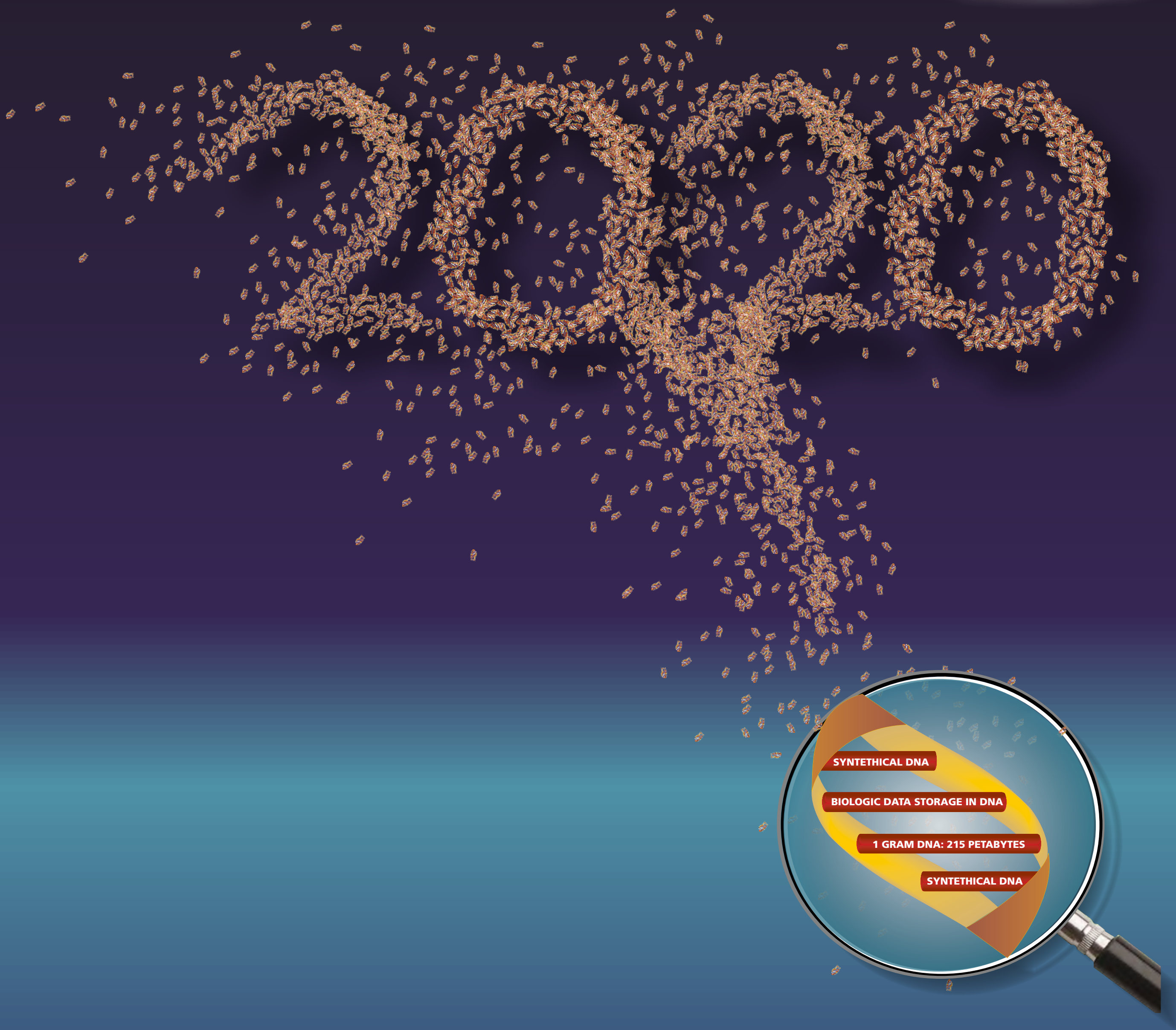
# BLAISE PASCAL MAGAZINE 84

Object Pascal / Internet / JavaScript / WebAssembly / Pas2Js / Databases
CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux

Blaise Pascal



Artificial DNA as the new HardDisk?
It will probably become the energy saviour of the world
The kbm ChangeTool for Lazarus in an updated version
and the new Lazarus Version 2.0.6 Extended
The new version 1.3 of TMSWebtools is available:
incredible new components
Docking Lazarus for your note book:
Undocking the IDE in Delphi
A new series: COMPASSIST: the activity indicator
and the notification center component
Lazarus: Moving the component palette
Delphi: the component palette refurbished

# BLAISE PASCAL MAGAZINE 84

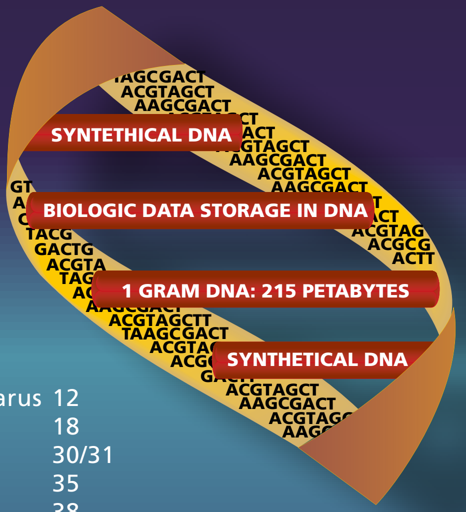
Object Pascal / Internet / JavaScript / WebAssembly / Pas2Js / Databases
CSS Styles / Progressive Web Apps
Android / IOS / Mac / Windows & Linux

Blaise Pascal

## CONTENT

### ARTICLES

Artificial DNA as the new HardDisk?
By Detlef Overbeek Page 6
It will probably become the energy saviour of the world...
The kbm ChangeTool for Lazarus in an updated version
By Detlef Overbeek Page 13
and the new Lazarus Version 2.0.6 Extended
The new version 1.3 of TMSWebtools is available:
incredible new components
By Bruno Fierens Page 19
Docking / Undocking Lazarus for your note book: Page 28
Undocking the IDE in Delphi Page 36
A new series: COMPASSIST: the activity indicator Page 40
and the notification center component Page 57
Lazarus: Moving the component palette Page 46
Delphi: the component palette refurbished Page 54



| | |
|---|---|
| What most people not know about Lazarus | 12 |
| C-Quel In need of Delphi support | 18 |
| Pascon Delphi & Lazarus Event | 30/31 |
| BPM Library | 35 |
| Lazarus 2.0.6 | 38 |
| Lazarus Free Special 2.0.6 | 39 |
| Lazarus Factory | 45 |
| Barnsten Delphi VCL Essentials Training | 53 |
| Delphi Company | 56 |
| Component 4 Dvelopers | 60 |

Publisher: PRO PASCAL FOUNDATION in collaboration © Stichting Ondersteuning Programmeertaal Pascal

Pascal is an imperative and procedural programming language, which Niklaus Wirth designed in 1968–69 and published in 1970, as a small, efficient language intended to encourage good programming practices using structured programming and data structuring. A derivative known as Object Pascal designed for object-oriented programming was developed in 1985. The language name was chosen to honour the Mathematician, Inventor of the first calculator: Blaise Pascal (see top right).

Niklaus Wirth

| | | |
|---|---|---|
| **Stephen Ball**<br>http://delphiaball.co.uk<br>@DelphiABall | | **Peter Bijlsma -Editor**<br>peter @ blaisepascal.eu |
| **Dmitry Boyarintsev**<br>dmitry.living @ gmail.com | **Michaël Van Canneyt,**<br>michael @ freepascal.org | **Marco Cantù**<br>www.marcocantu.com<br>marco.cantu @ gmail.com |
| **David Dirkse**<br>www.davdata.nl<br>E-mail: David @ davdata.nl | **Benno Evers**<br>b.evers<br>@ everscustomtechnology.nl | **Bruno Fierens**<br>www.tmssoftware.com<br>bruno.fierens @ tmssoftware.com |
| **Holger Flick**<br>holger @ flixments.com | | |
| **Primož Gabrijelčič**<br>www.primoz @ gabrijelcic.org | **Mattias Gärtner**<br>nc-gaertnma@netcologne.de | **Peter Johnson**<br>http://delphidabbler.com<br>delphidabbler @ gmail.com |
| **Max Kleiner**<br>www.softwareschule.ch<br>max @ kleiner.com | **John Kuiper**<br>john_kuiper @ kpnmail.nl | **Wagner R. Landgraf**<br>wagner @ tmssoftware.com |
| **Vsevolod Leonov**<br>vsevolod.leonov@mail.ru | | **Andrea Magni** www.andreamagni.eu<br>andrea.magni @ gmail.com<br>www.andreamagni.eu/wp |
| | **Paul Nauta PLM Solution** Architect<br>CyberNautics<br>paul.nauta @ cybernautics.nl | **Kim Madsen**<br>www.component4developers |
| **Boian Mitov**<br>mitov @ mitov.com | | **Jeremy North**<br>jeremy.north @ gmail.com |
| **Detlef Overbeek - Editor in Chief**<br>www.blaisepascal.eu<br>editor @ blaisepascal.eu | **Howard Page Clark**<br>hdpc @ talktalk.net | **Heiko Rompel**<br>info @ rompelsoft.de |
| **Wim Van Ingen Schenau -Editor**<br>wisone @ xs4all.nl | **Peter van der Sman**<br>sman @ prisman.nl | **Rik Smit**<br>rik @ blaisepascal.eu |
| **Bob Swart**<br>www.eBob42.com<br>Bob @ eBob42.com | **B.J. Rao**<br>contact @ intricad.com | **Daniele Teti**<br>www.danieleteti.it<br>d.teti @ bittime.it |
| **Anton Vogelaar**<br>ajv @ vogelaar-electronics.com | **Robert Welland**<br>support @ objectpascal.org | **Siegfried Zuhr**<br>siegfried @ zuhr.nl |

Editor - in - chief
Detlef D. Overbeek, Netherlands Tel.: Mobile: +31 (0)6 21.23.62.68
News and Press Releases email only to editor@blaisepascal.eu

**Editors**
Peter Bijlsma, W. (Wim) van Ingen Schenau, Rik Smit

**Correctors**
Howard Page-Clark, Peter Bijlsma

**Trademarks** All trademarks used are acknowledged as the property of their respective owners.

**Caveat** Whilst we endeavour to ensure that what is published in the magazine is correct, we cannot accept responsibility for any errors or omissions. If you notice something which may be incorrect, please contact the Editor and we will publish a correction where relevant.

**Subscriptions** ( 2019 prices )

| | Internat. excl. VAT | Internat. incl. 9% VAT | Shipment |
|---|---|---|---|
| **Printed Issue ±60 pages** | € 250 | € 261,60 | € 85,00 |
| **Electronic Download Issue 60 pages** | € 60 | € 65,40 | ——— |
| **Printed Issue inside Holland (Netherlands) ±60 pages** | ——— | € 200,00 | € 60,00 |

Member and donator of WIKIPEDIA

Subscriptions can be taken out online at www.blaisepascal.eu or by written order, or by sending an email to office@blaisepascal.eu
Subscriptions can start at any date. All issues published in the calendar year of the subscription will be sent as well.
**Subscriptions run 365 days.** Subscriptions will not be prolonged without notice. Receipt of payment will be sent by email.
Subscriptions can be paid by sending the payment to:
**ABN AMRO Bank Account no. 44 19 60 863** or by credit card or Paypal
Name: Pro Pascal Foundation-Foundation for Supporting the Pascal Programming Language (Stichting Ondersteuning Programeertaal Pascal)
**IBAN: NL82 ABNA 0441960863 BIC ABNANL2A VAT no.: 81 42 54 147** (Stichting Programmeertaal Pascal)

**Subscription department**
Edelstenenbaan 21 / 3402 XA IJsselstein, The Netherlands
Mobile: + 31 (0) 6 21.23.62.68 office@blaisepascal.eu

## Copyright notice

# From your editor 1

We are living in very exciting and challenging times, even if they are not always good.
I have accumulated enough life experience to reckon that history swings rather like a pendulum with an average period of about thirty years.
If you consider the recent past from the end of the 19th century to the present day, you can discern a pattern such as this:

1900s
The turn of the 19th century, leading up to the First World War, followed by the costly failure of communism in some countries, and the emergence of super-powers. The West's increasing reliance on crude oil making supplier countries in the Middle East more economically dominant.

1930s
The Thirties showing increasingly the signs of the upcoming Second World War.

1960s
The Sixties: a period of flower power, Women's Lib, baby boomers coming to power, increasing freedom of sexual expression. A new politics based on burgeoning post-war economic growth.

1990s
The Nineties was a period with many strikes and life-changing events such as the fall of the Berlin Wall, the end of the cold war, and the beginning of so-called super-capitalism. Notable leaders such as the Iron Lady, the little lady with the handbag. Towards the close of this period most people see that the earth is approaching an environmental disaster, burning up its resources, and world leaders must unite to mitigate the causes and consequences.

2020s
A new 2020s era, possibly demanding we make the biggest changes we have ever needed to undertake: coming to terms with super-capitalism; global warming leading to a struggle for life in many places; the emergence of new viruses and drug-resistant germs; a battle against pollution; increasing numbers of people struggling to escape poverty and find employment.

I have highlighted the downsides of these periods. You may notice three very remarkable features:

1 - we have always caused the problems ourselves
2 - we have always found a solution
3 - we have (so far) always survived

My history teacher long ago taught me that a society which is unable to recognise or solve its main problems is headed for collapse.
Ancient Greek, ancient Roman and ancient Chinese civilisations are primary examples. Their failures were not failures of one or two leaders, but the failure of a whole system.

Why the history lesson?
My point is this: Embarcadero has a problem.
For the second time recently the company is acting as if it does not understand its community.
The primary goal of making a large profit has blinded them: by dismissing all their technicians (their biggest resource!) they may make more money in the short term, but at the cost of losing those who understand Delphi, and are able to maintain it, and most importantly able to create an innovative future for Delphi..

# From your editor 2

A new company (Idera) took ownership, which made us hope for a bright future… but we discover unfortunately that while they are good at managing money, they seem to lack elementary understanding of how to create and build cutting-edge software tools, and also lack the skill needed to take on board user group feedback.

I learned long ago that a user group is like family. If you don't treat your family with kindness and respect – listening, and letting the family help you – if your relationship disintegrates, then you will lose your family. Then what do you have?
I do not want to go on grouching. But I want to sound a warning, so let's try to find a way beyond this situation. I want Delphi to be great again (where did I first hear that?) – at one time it created a critical turning point for developers. It was a completely new way of thinking.
We seem to have lost that.

I want to stress that we All need to work on these failures, and I think we can.
If Embarcadero were to take us seriously as a user group, we could find a new élan.
Because Embarcadero is not helping presently, we need to create our own initiatives, producing our own good ideas about what needs to be developed for Delphi.
Perhaps Embarcadero will follow that.

One idea is to create an interface that is capable of helping teenage developers with little experience.
We need such as tomorrow's developers, otherwise we will be working only with a niche product of limited lifespan. For now the internet is the future. How could you not develop for that?

A downside to Embarcadero-Idera's acquisition of further companies is the lack of interaction between these new entities.
Developers need to know and understand these potential resources, so we can use them.
We need to help each other.

These woes have been with us for some time.
Happily there is another Pascal development environment:
Lazarus with FPC.
The time is long gone when people could legitimately ask: "Oh Lazarus, does that work?"
Lazarus development is presently so good there is almost nothing that Delphi does that Lazarus cannot also do. In some areas, Lazarus is ahead of Delphi functionality.

Why is this?
It has a very strong user group, and the developers who use it care about it.
Lazarus matters to them.
This is how the market should work – to make both Delphi and Lazarus stronger and better.
People are free to choose between them, or to use both.

Long live Pascal.
And you!
Let's have a lot of creative fun in the coming year.
I'm sure it will have its surprises.

Detlef Overbeek

# ARTIFICIAL DNA AS THE NEW HARD DISK ?

Is it possible to store data using synthetic **DNA as a storage medium** – saving bits and bytes in a molecule?
Modern data storage technology can hardly keep up with the ever-increasing demand for data storage. Pioneers in a campus based in the English village of Hinxton are working on the development of synthetic DNA as a data storage solution.
**Is this safe?** Does it really work? **Could it contaminate our own DNA?**
**What advantage does DNA have over other storage media?**
Some people claim that DNA has the potential to store such vast amounts of data that hard disks may soon be obsolete. Are they correct?
Current hard disk technology uses a great deal of energy (a significant percentage of worldwide energy use is already devoted to data storage and retrieval), and consume much of the world's scarce precious metals. Could DNA ever be cost-effective as a replacement?

**A recent Nature article included the following:**
Upload your latest holiday photos to Facebook, and there's a chance they'll end up stored in Prineville, Oregon, a small town where the firm has built three giant data centres and is planning two more. Inside these vast factories, bigger than aircraft carriers, tens of thousands of circuit boards are racked row upon row, stretching down windowless halls so long that staff ride through the corridors on scooters.

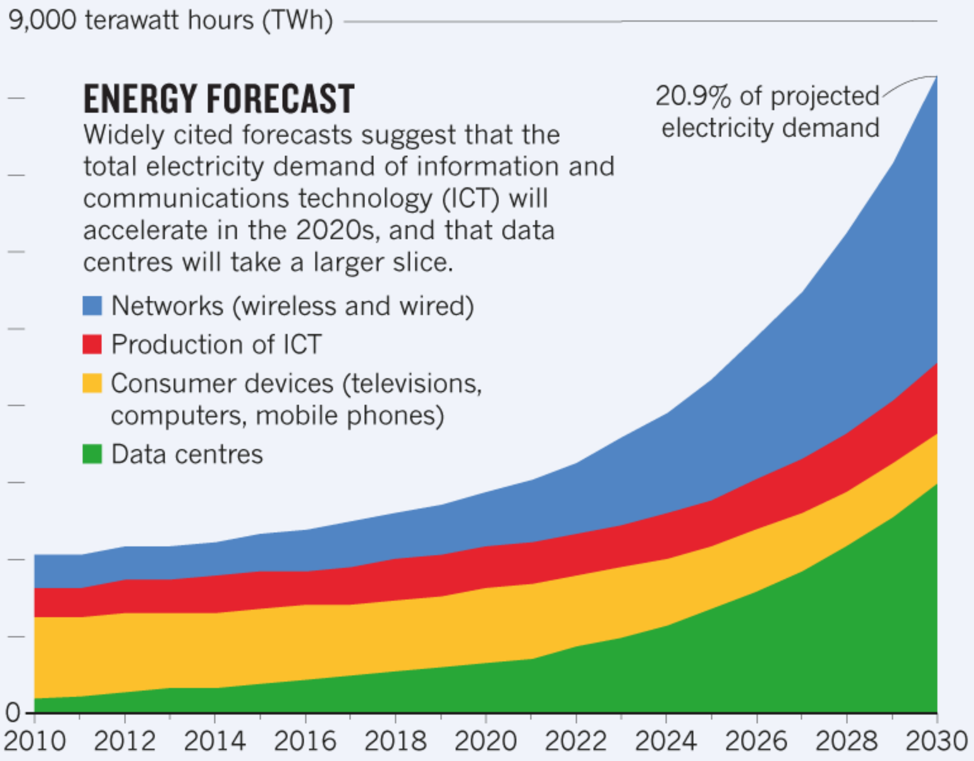
These huge buildings are the treasuries of the new industrial kings: the information traders.
The five biggest global companies by market capitalization this year are currently **Apple, Amazon, Alphabet, Microsoft** and **Facebook,** replacing titans such as **Shell** and **ExxonMobil.**

Although information factories might not spew out black smoke or grind greasy cogs, they are not bereft of environmental impact.
As demand for Internet and mobile-phone traffic skyrockets, the information industry could lead to an explosion in energy use. See the Energy Forecast illustration on the next page.

**FORECAST:**
Already, data centres use an estimated 200 terawatt hours (TWh) each year. That is more than the national energy consumption of some countries, including Iran, but half of the electricity used for transport worldwide, and just 1% of global electricity demand (see Global electricity demand on the next page. Data centres contribute around 0.3% to overall carbon emissions, whereas the information and communications technology (ICT) ecosystem as a whole — under a sweeping definition that encompasses personal digital devices, mobile-phone networks and televisions — accounts for more than 2% of global emissions. That puts ICT's carbon footprint on a par with the aviation industry's emissions from fuel. What could happen in the future is hard to forecast. **But one of the most worrying models predicts that electricity use by ICT could exceed 20% of the global total by the time a child born today reaches her teens**, with data centres using more than one-third of that (see Energy Forecast on the next page).

If Bitcoin, the computationally intensive crypto currency, continues to grow, a sharp rise in energy demand could come sooner rather than later.
For now, despite rising demand for data, ICT's electricity consumption is staying nearly flat, as increased Internet traffic and data loads are countered by increased efficiencies - including shuttering older facilities in favour of ultra-efficient centres such as Prineville's.

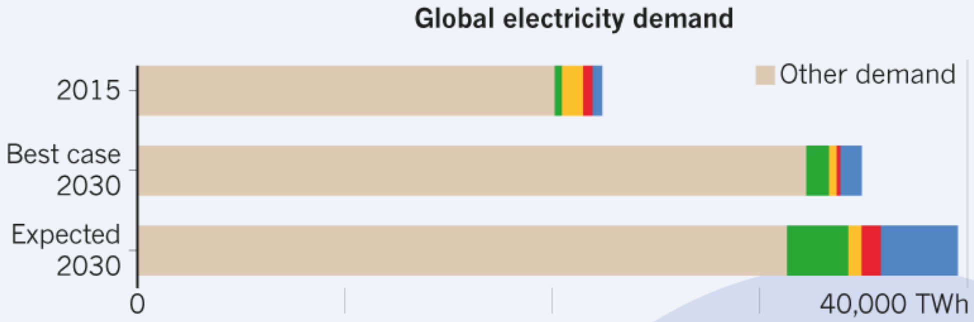
But those easy wins could end within a decade. *"The trend is good right now, but it's questionable what it's going to look like in 5–10 years,"* says Dale Sartor, who oversees the Center of Expertise for Energy Efficiency in Data Centers at the US Department of Energy's Lawrence Berkeley National Laboratory in Berkeley, California.



## ENERGY FORECAST

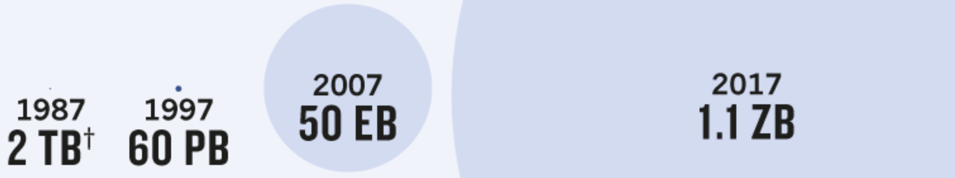
Widely cited forecasts suggest that the total electricity demand of information and communications technology (ICT) will accelerate in the 2020s, and that data centres will take a larger slice.

The chart above is an 'expected case' projection from Anders Andrae, a specialist in sustainable ICT. In his 'best case' scenario, ICT grows to only 8% of total electricity demand by 2030, rather than to 21%.



## INTERNET EXPLOSION

Internet traffic* is growing exponentially, and reached more than a zettabyte (ZB, $1 \times 10^{21}$ bytes) in 2017.



*Traffic to and from data centres.
†TB, terabyte ($10^{12}$ bytes); PB, petabyte ($10^{15}$ bytes); EB, exabyte ($10^{18}$ bytes).

©nature

### ICT'S ENERGY VIGILANTLY MANAGED

With the spectre of an energy-hungry future looming, scientists in academic labs and engineers at some of the world's wealthiest companies are exploring ways to keep the industry's environmental impact in check.
They are streamlining computing processes, switching to renewables and investigating better ways to cool data centres and to recycle their waste heat. **ICT's** energy use must be *" vigilantly managed"*, says **Eric Masanet**, an engineer at **Northwestern University in Evanston**, Illinois, who co-authored an **International Energy Agency** (IEA) report last year on digitalization and energy. *"If we stay on top of it"*, he says, *"we should keep future energy demand in check"*.

**SERENDIPITY** (the faculty of making beneficial discoveries by accident) is often a factor in the creation of important inventions. Using DNA for large-scale information storage is such a case: the result of a happy coincidence that could turn the way we store data, and perhaps even the way we live, completely upside down.
The **Wellcome Genome Campus** in the English village of Hinxton is home to the **European Bioinformatics Institute (EBI),** part of the **European Molecular Biology Laboratory (EMBL**).
The DNA serendipity is associated with this international scientific research organization, which employs about six hundred people who conduct research and provide services in the field of biological data. The entire code of the human genome was mapped here.
The **Wellcome Trust Sanger Institute**, also on the campus, is currently carrying out its ambitious **Darwin Tree of Life** project to map the genomes of all living organisms. 66,000 species have so far been mapped in the United Kingdom.
**`https://www.ebi.ac.uk/`**
One principle undergirding such academic institutions is that biological data should be freely available to scientists around the world: science should be open. The open-access EBI is therefore crucial for scientific research, receiving two million requests for information each year. A similar amount of biological data is readily available elsewhere only in the United States.

When the American government ecently experienced a shut-down, Hinxton saw a sharp increase in the number of requests for data.

At the time when the entire human genome was being mapped at the Genome Campus in Hinxton, two scientists from the EBI, **Nick Goldman** and **Ewan Birney**, attended a conference in Hamburg on the ever-expanding flow of genetic information. One evening they were discussing their worries.
*"We were expected to store the rapidly increasing amount of genetic information we collected at the same speed, and make it accessible to scientists around the world. That was a problem.Our computers were indeed better, but not so much better that* o*they could keep up with those explosive information flows. And our funding also barely increased."*

**Nick Goldman explains DNA storage. Credit: UNICEF**

Scientists the United States, who were also mapping the human genome in a parallel project, had indicated a year earlier that they might have to stop recording the data, because they were unable to save the rapidly increasing amounts that accumulated.
Although the **European Bioinformatics Institute** received 'generous funding' from the 27 countries that had established the Institute by treaty, and although the UK as host contributed millions as well, the scientists needed more money.
They were asking for a further £26 million. The storage problem was getting steadily worse.

*“The nature of our work requires that we see DNA molecules as a carrier of information. After all, what we record here are the sequences - the codes of the genome - of the DNA molecules. Such a sequence looks no different than a computer file with four-letter codes.*

*First you take a piece of human tissue. You extract the DNA from it with a centrifuge. You break that into smaller pieces. Finally, there is a machine that reads every building block of every piece of DNA and sends it back to you in a computer file.*

*Eventually you then have a computer file with the order of those different building blocks: indicated by the letters A, G, C and T. You can read combinations of these four letters in groups of three. Ultimately, with all those little pieces of DNA, you have to make a huge jigsaw puzzle and you get 23 pieces of DNA, the chromosomes.”*

That is how the **human genome**, the genetic blueprint for humans, is recorded. The moment of serendipity arrived:

Would it not be possible to convert digital information consisting of codes of zeros and ones into DNA sequences, encoded using the four letters A, G, C and T? Could you not store information exactly as genes store information? What started as an off-beat idea soon became a serious pursuit.

**Could you not store information exactly as genes store information?**

Information storage encoded as zeros and ones on a magnetic disk could correspond to storage of information encoded using the four letters A, G, C and T in a molecule. This is not a far-fetched correspondence. *“We had all the mathematical skills to design a system that would convert digital information into information on a DNA molecule and convert it back, but we didn't have a lab where we could make the synthetic DNA, although we knew the people who could.”*

They made a few calculations, recorded in a paper stating that it is in principle possible to convert digital information into information in a DNA molecule. The system they designed actually consisted of a software program that tells the computer how the code of zeros and ones should be converted to the four letters A, G, C, and T in the DNA. They sent the codes that they received to colleagues who could synthesize DNA molecules in their lab. Soon afterwards their colleagues returned a tube with synthetic DNA encoding that information.

What they encoded in that first DNA test included:

**Shakespeare's 154 sonnets, a clip of Martin Luther King's famous I Have a Dream speech, a pdf of James Watson’s and Francis Crick’s original publication about discovering the structure of DNA, and a photo of the old oak on the Genome Campus site.**

They supplied the software needed to convert all this information into DNA code. Everything was transferred flawlessly.

How digital data gets stored in DNA

To encode the document into archival storage copies in DNA, first, the digital files were converted from the binary code using 0s and 1s of digital data into sequences of A, C, T and G by EMBL-EBI. Twist Bioscience then synthesized the DNA in short segments in the sequence order provided. The short DNA segments each contain about 20 bytes of data as well as a sequence number to indicate their place within the overall sequence.

This is the process of storage. To ensure that the file is stored accurately, Twist reads the sequence back to ensure 100 percent accuracy. Finally, to store the document for the future, Imagene SA placed copies of the document **encoded in DNA into DNAshells, small stainless-steel capsules, laser sealed under inert atmosphere, which helps preserve the DNA for up to thousands of years if kept at room temperature.**

This first case of information storage in DNA took place in 2013. But although **in July 2019 the Scientific American classed DNA information storage as one of the ten most promising technological breakthroughs**, we are not yet storing our data in DNA molecules, because so much energy is needed to extract, search for or change that information.

Meantime, the amount of information we store is increasing at a dizzying pace. In the past few years, we have stored more data than in the entire previous period of human history.

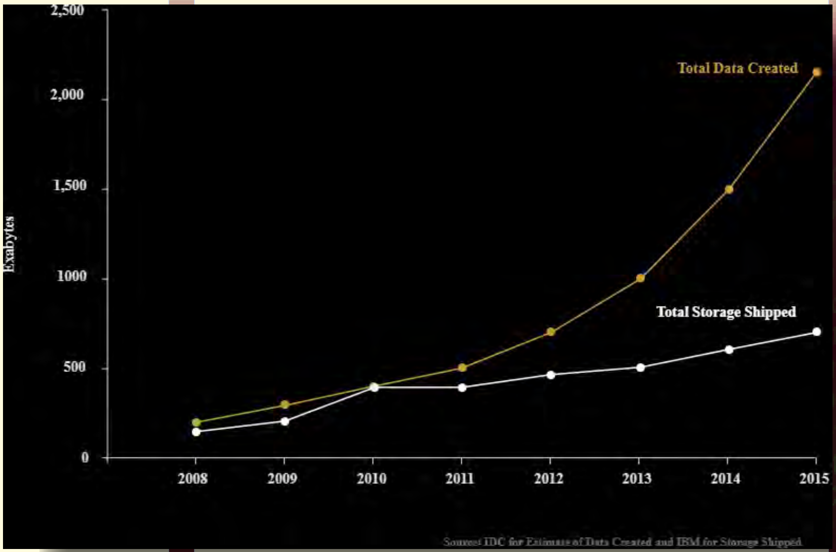
Current technology hardly meets any good standard of sustainable storage. Not only is the amount of data increasing with unprecedented speed, but the media used to store all that data is constantly changing, as are the systems used to read that data.

**Dina Zielinski**, a scientist at the Institut Curie in Paris involved in DNA data storage, explains the problem. *“All technology is eventually lost or becomes obsolete; and before that happens the data must transferred to new technology. You have to do this in good time.*
*Even the bits of digital technology perish over time.*
**We refer to this as ‘bit rot”.**

The fact that we can read genetic information from fossils proves how sustainable DNA is. If you store DNA in a cool environment it lasts unchanged for thousands of years.
If you were to put a synthetic DNA molecule in your body, it would probably be broken down.
The synthetic DNA molecule is not encoded with protein-building commands. **Synthetic DNA deliberately includes stop-building codes. It is almost impossible that you would get contaminated with this artificial DNA.**
Reading DNA codes and a lot of writing the codes is still too expensive and too slow. Even if you scaled it up efficiently, it is still too slow.
We must go at least ten to the power five or six faster before this technology can become viable. That sounds pretty hopeless, except that advances in reading genome sequences over the first five-year period did upscale by ten to the power of six. Likewise consider the first computers which occupied entire buildings. These days we use them scaled to the size of a watch.

Several researchers are working on designing better algorithms that make the margin of error in transferring from digital to DNA (and back) smaller, and the entire process faster. Yaniv Erlich and Dina Zielinski, for example, developed a new algorithm called DNAfountain. Using this, they were able to encode 2.14 megabytes of computer data, including a computer operating system and a gift card from Amazon, and read it again, error-free in DNA. Their strategy set a new a record for data density: 215 petabytes in one gram of DNA (a petabyte or PB is 1015 bytes of digital information).
Experience shows that upscaling can go fast and DNA data storage is therefore *“commercially sensitive technology”* that several companies claim they can already do. However, most companies are not very keen on sharing their latest insights.

The big breakthroughs will probably come from the commercial corner, since that is where the big money for funding development lies. Technology that has been developed initially with public money may be exploited commercially by technology companies.



Source: IDC for Estimates of Data Created and IBM for Storage Shipped

**Total data creation compared  with total storage shipped**

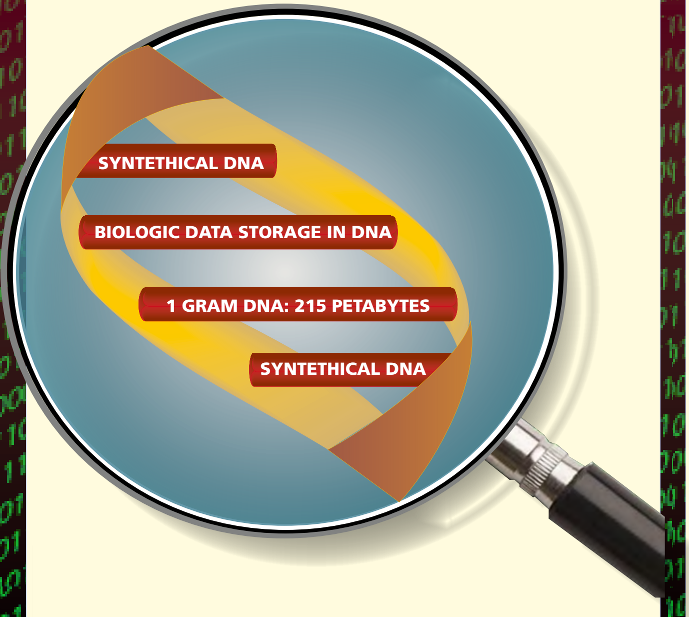
For example, Microsoft recently introduced this:
**`https://github.com/microsoft/Microsoft.Shared.Dna.Json/blob/master/Microsoft.Shared.Dna.Json.Profile/JsonDnaPerformanceTests.cs`**

Other companies are also making substantial investments in the further development of this technology. The market leader is the Californian company Twist Bioscience, **`https://www.twistbioscience.com/products/storage`** a specialist in the synthesis of DNA. The company’s CEO says *“Our goal is to reduce the price of DNA storage to one hundred dollars per terabyte. Which is the same price as for storage on a hard drive, but with all the benefits of DNA: durability, robustness and density!”*

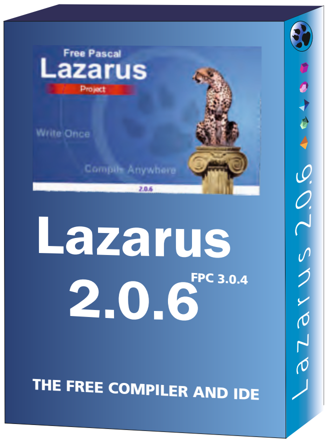
**To celebrate the 30-year anniversary of the United Nations Convention on the Rights of the Child, UNICEF found a way to ensure that this important document is stored for the foreseeable future.**
**The UN Convention on the Rights of the Child (CRC) was unanimously adopted by the General Assembly of the United Nations on November 20th, 1989. In 2019, the document has been stored in synthetic DNA. The capsule containing the DNA will be kept in one of the safest places in the world, in the permafrost at the Arctic World Archive in Svalbard, Norway.**

UNICEF Norway's Executive Director Camilla Viken says:
*"The Convention on the Rights of the Child is one of the world's most important documents. It protects our children and their rights and will now be a part of our organisational DNA. The Children's Rights Convention is the first document to be stored officially in DNA."*
12 November 2019, Cambridge, Oslo, Pessac, San Francisco

Because **Twist** has technological knowledge in both programming (converting digital to DNA codes) and synthesizing (writing the codes in DNA), the company has everything it needs to further develop the technology considerably.
The most expensive aspect is writing the codes in the DNA and it is there that Twist has unique and valuable knowledge.
**https://www.youtube.com/watch?v=v_9Bt8sBuaY#action=share**
Exactly what knowledge, the company leaves aside because of its commercial sensitivity, but, says **Emily Leproust, PhD (Chief Executive Officer)**:
'We work with a timeline of three to five years.'



The Lazarus Factory



# WHAT MOST PEOPLE REALLY DO NOT KNOW ABOUT LAZARUS:

IT'S NATIVE TO

**WINDOWS**

**LINUX**

**APPLE (MAC)**

**RASPBERRY**

## BUILD ONCE
## COMPILE ANYWHERE

# THE UPDATED VERSION OF THE KBM MEMTABLE CHANGETOOL
## PROJECT OF THE PRE-INSTALLED SPECIAL VERSION OF LAZARUS 2.0.6

starter expert

Because we were not yet satisfied about the last version of the **ClientDataSet kbmMemTable** for Lazarus we wanted to update that version to a much easier to understand and use program.

At this moment this is ready-made for the free **Special Version for Lazarus 2.0.6.** The Special Lazarus you can download for free from our website – it has integrated the Webcore Suite from TMSSoftware and also the kbmStandard version MemTable among quite a lot of other components. So you do not need to install any extra's, what you need to do is get your free license from TMSSoftware and install that for Lazarus. If you have trouble with that please let me know.

The address is `https://www.tmssoftware.com/site/tmswebcore.asp` and here you can download your Webcore for Lazarus and even Delphi. The special Lazarus is available for everybody: `https://www.blaisepascalmagazine.eu/9372-2/` You can compile and add other components to this version. Do **NOT** do: CleanUp and Build from the Lazarus Menu. Then you will damage the files for this version, because it can NOT recompile the sources. But you probably will never have to…If, then install that special version again. Be sure to have a copy of this on your system Lazarus2.0.6
You can unpack this zip file and simply copy it to any directory, even a USB stick and it will work.
You need to install the TMS Webcore Trial: TMSWEBCoreXE12_BIN.zip
You can install only Webcore for Lazarus or install it as well for Delphi.

### LOAD EXISTING:

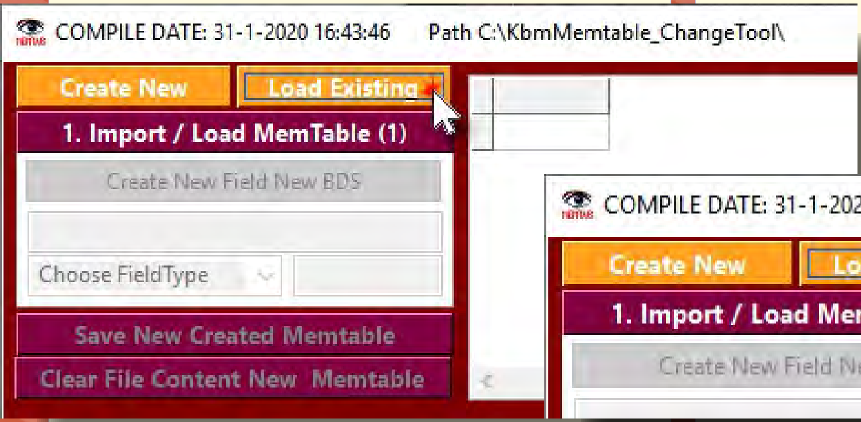

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_ChangeTool\
Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable

Figure 1: Load an existing Table enables the next step

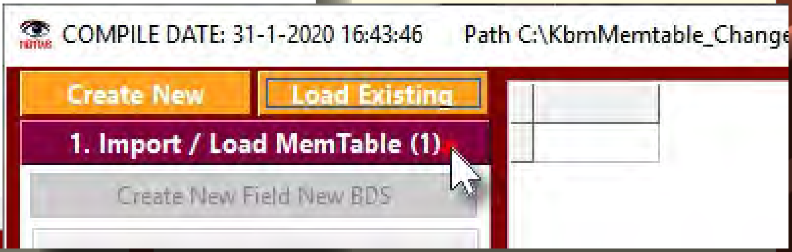
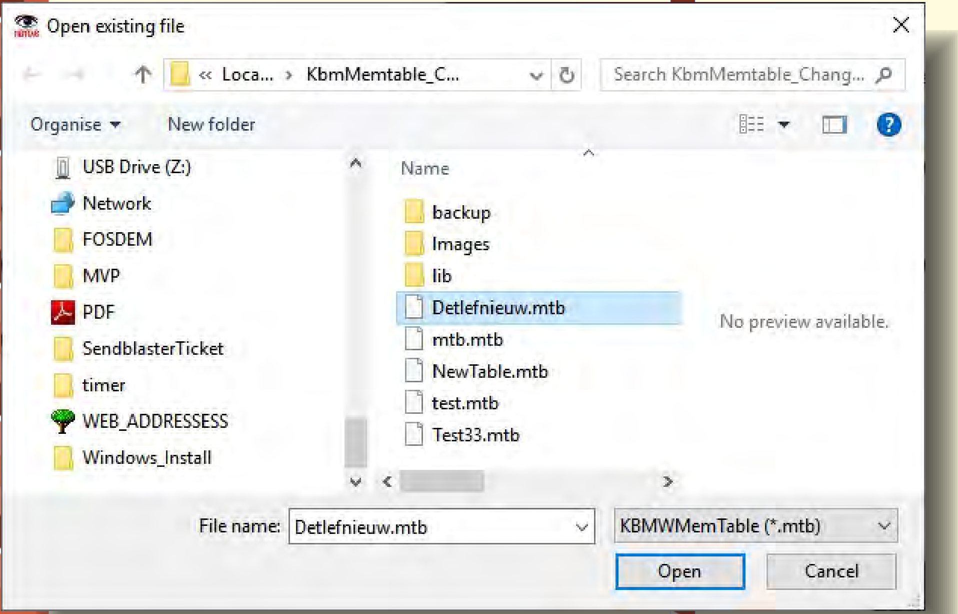

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_Change

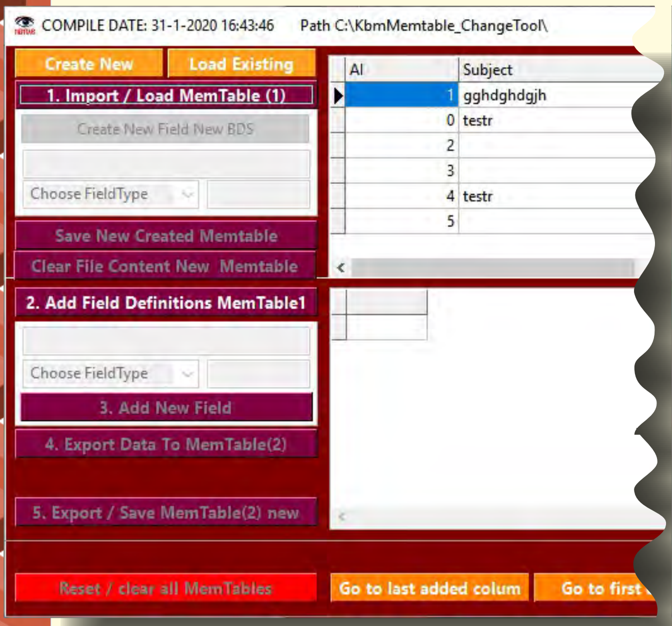
Figure 2: Import / Load MemTable



File name: Detlefnieuw.mtb
KBMWMemTable (*.mtb)

Figure 3: Open existing File

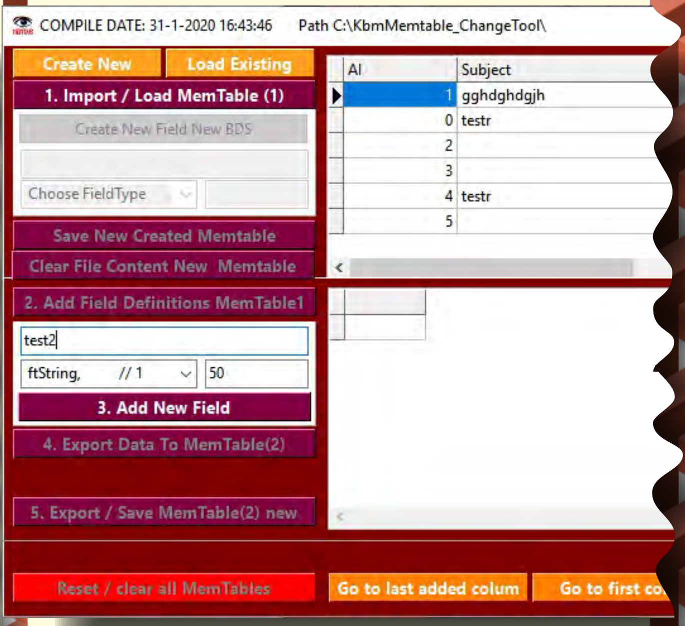
## LOAD EXISTING:


Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable
2. Add Field Definitions MemTable1
Choose FieldType
3. Add New Field
4. Export Data To MemTable(2)
5. Export / Save MemTable(2) new
Reset / clear all MemTables
Go to last added colum
Go to first

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_ChangeTool\

| AI | Subject |
|---|---|
| 1 | gghdghdgjh |
| 0 | testr |
| 2 | |
| 3 | |
| 4 | testr |
| 5 | |

Figure 4: Step 1 -Import / Load Memtable; Step2 becomes available


Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable
2. Add Field Definitions MemTable1
3. Add New Field
4. Export Data To MemTable(2)
5. Export / Save MemTable(2) new
Reset / clear all MemTables
Go to last added colum
Go to first co

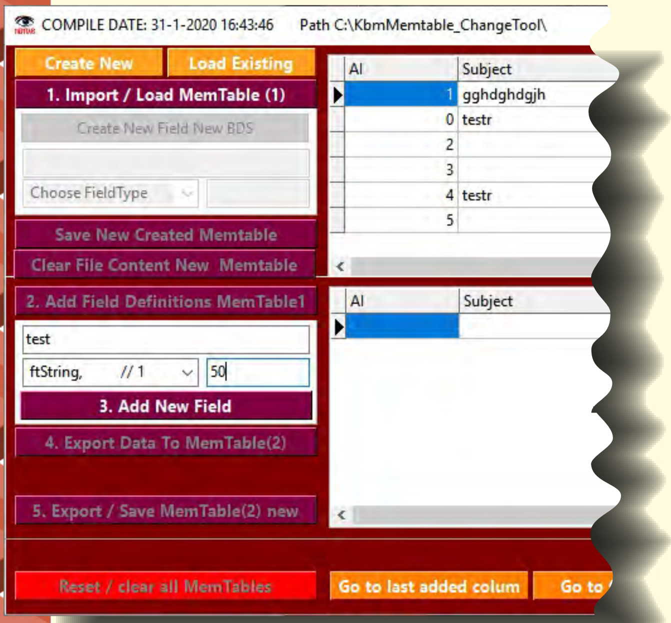
COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_ChangeTool\

| AI | Subject |
|---|---|
| 1 | gghdghdgjh |
| 0 | testr |
| 2 | |
| 3 | |
| 4 | testr |
| 5 | |

test2

ftString, // 1 50

Figure 5: Now you can click Add New Field: and add the properties into thos 3 fields.


Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable
2. Add Field Definitions MemTable1
3. Add New Field
4. Export Data To MemTable(2)
5. Export / Save MemTable(2) new
Reset / clear all MemTables
Go to last added colum
Go to

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_ChangeTool\

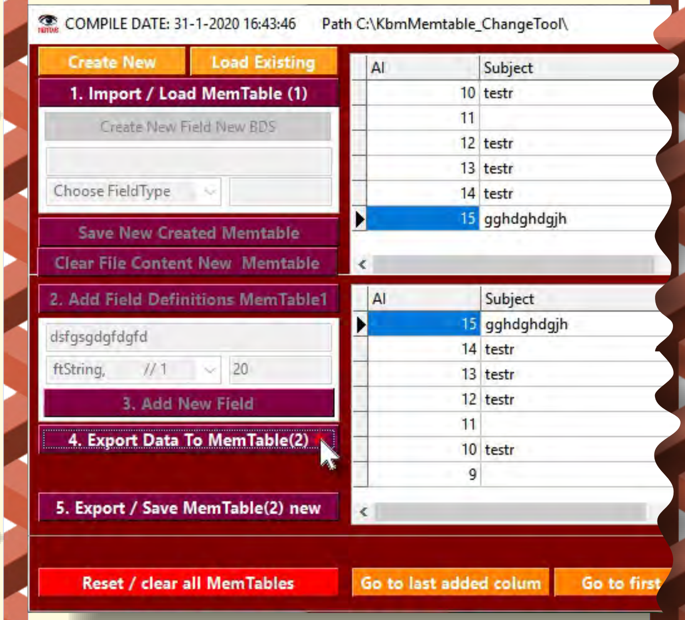
| AI | Subject |
|---|---|
| 1 | gghdghdgjh |
| 0 | testr |
| 2 | |
| 3 | |
| 4 | testr |
| 5 | |

test

ftString, // 1 50

| AI | Subject |
|---|---|
| | |

Figure 6: Having done that enables the button Add New Filed


Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable
2. Add Field Definitions MemTable1
3. Add New Field
4. Export Data To MemTable(2)
5. Export / Save MemTable(2) new
Reset / clear all MemTables
Go to last added colum
Go to first

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_ChangeTool\

| AI | Subject |
|---|---|
| 10 | testr |
| 11 | |
| 12 | testr |
| 13 | testr |
| 14 | testr |
| 15 | gghdghdgjh |

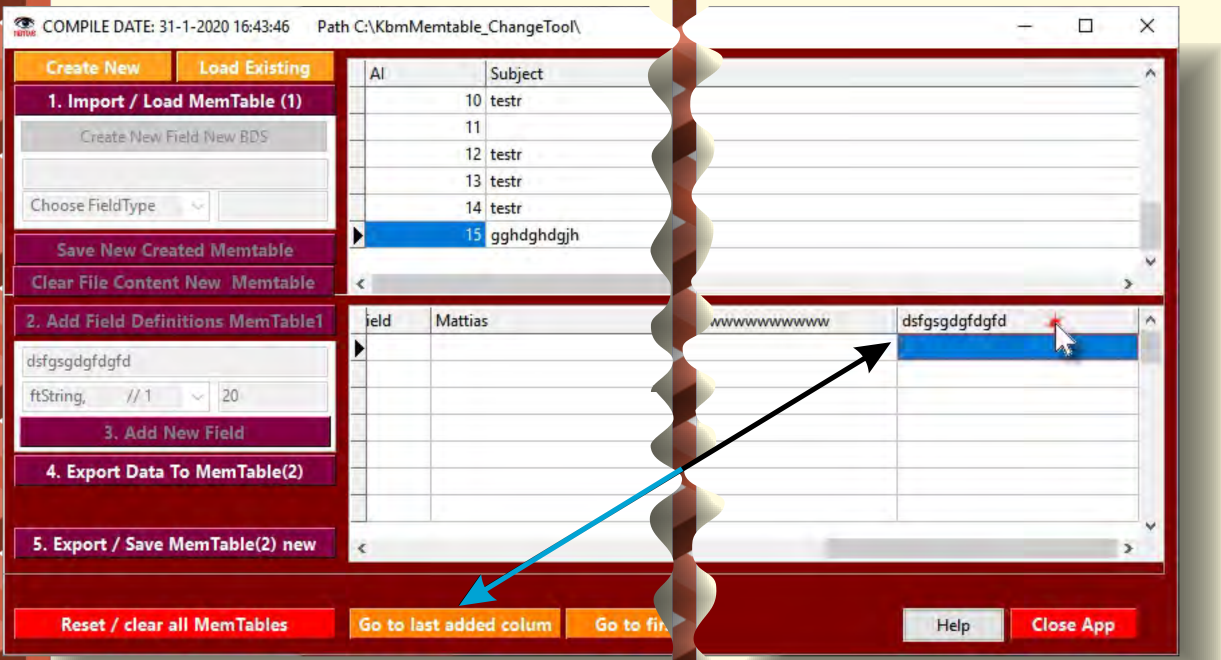
dsfgsgdgfdgfd

ftString, // 1 20

| AI | Subject |
|---|---|
| 15 | gghdghdgjh |
| 14 | testr |
| 13 | testr |
| 12 | testr |
| 11 | |
| 10 | testr |
| 9 | |

Figure 7: The export of all the data becomes available

## LOAD EXISTING:


COMPILE DATE: 31-1-2020 16:43:46
Path C:\KbmMemtable_ChangeTool\
Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable
2. Add Field Definitions MemTable1
3. Add New Field
4. Export Data To MemTable(2)
5. Export / Save MemTable(2) new
Reset / clear all MemTables
Go to last added colum
Go to fi
Help
Close App

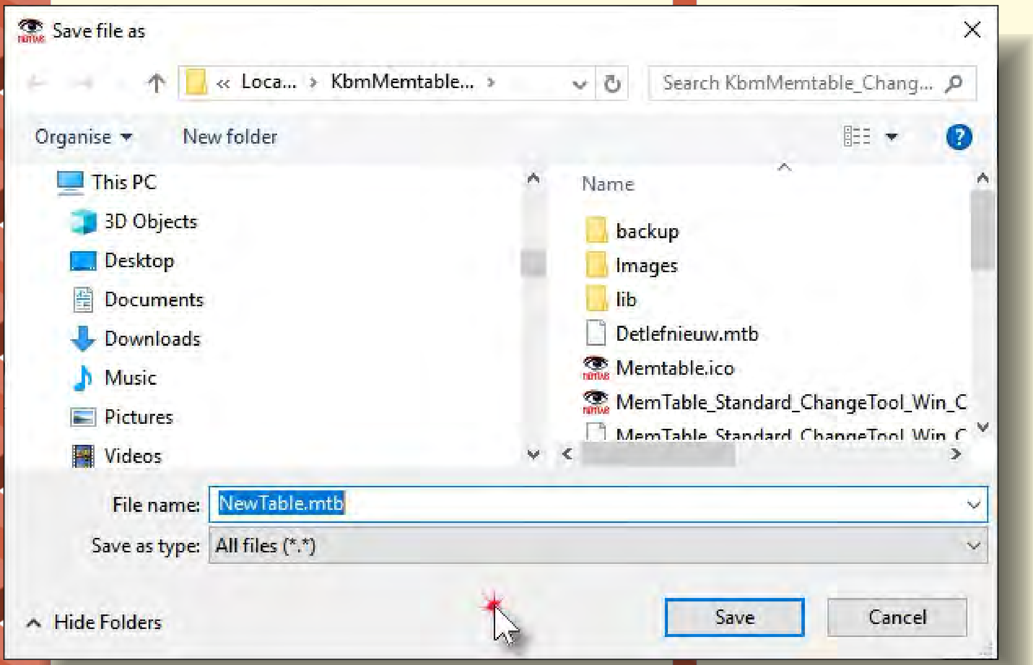
| AI | Subject |
|---|---|
| 10 | testr |
| 11 | |
| 12 | testr |
| 13 | testr |
| 14 | testr |
| 15 | gghdghdgjh |

dsfgsgdgfdgfd

ftString, // 1 — 20

ield | Mattias | wwwwwwwwwww | dsfgsgdgfdgfd

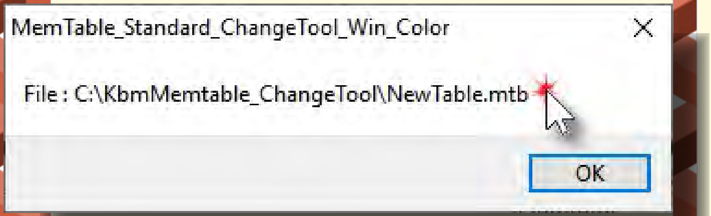
Figure 8: Push the button go to the last edited column shows the field at once.



File name: NewTable.mtb

Save as type: All files (*.*)

Figure 9: Save the Changed Table

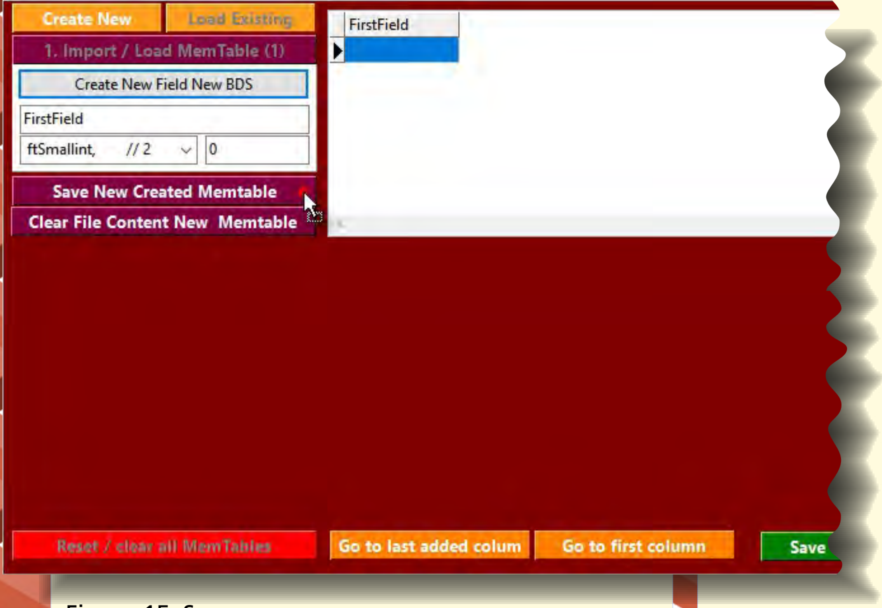


File : C:\KbmMemtable_ChangeTool\NewTable.mtb

Figure 10: Confirmation.
This is the last step in the Load Existing branche.

Now the Create New Branche starts →
on the next Page

## CREATE NEW

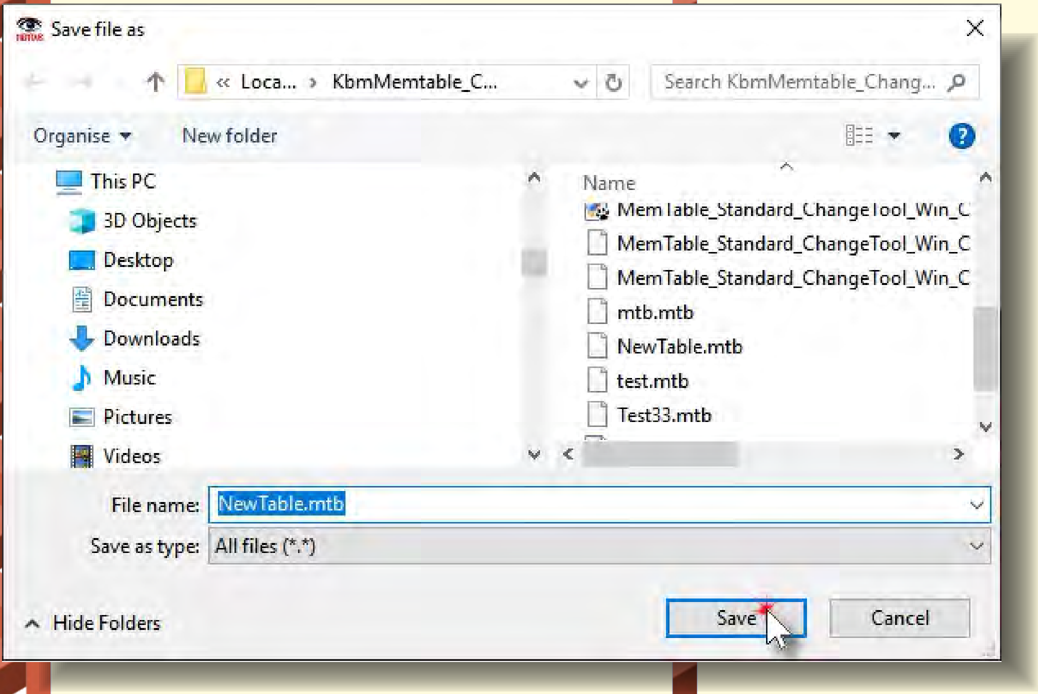

Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Save New Created Memtable
Clear File Content New Memtable
Reset / clear all MemTables
Go to last added colum
Go to first column
Save

FirstField

ftSmallint, // 2 | 0

FirstField

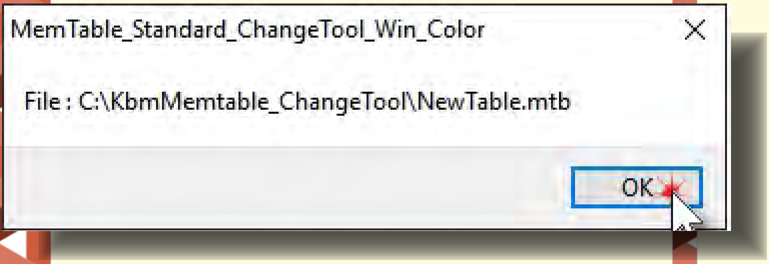
Figure 15: Save


Save file as
« Loca... › KbmMemtable_C...
Search KbmMemtable_Chang...
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Name
MemTable_Standard_ChangeTool_Win_C
MemTable_Standard_ChangeTool_Win_C
MemTable_Standard_ChangeTool_Win_C
mtb.mtb
NewTable.mtb
test.mtb
Test33.mtb
Hide Folders
Save
Cancel

File name: NewTable.mtb

Save as type: All files (*.*)

Figure 16: Naming



MemTable_Standard_ChangeTool_Win_Color

File : C:\KbmMemtable_ChangeTool\NewTable.mtb

Figure 17: Confirmation

## CREATE NEW

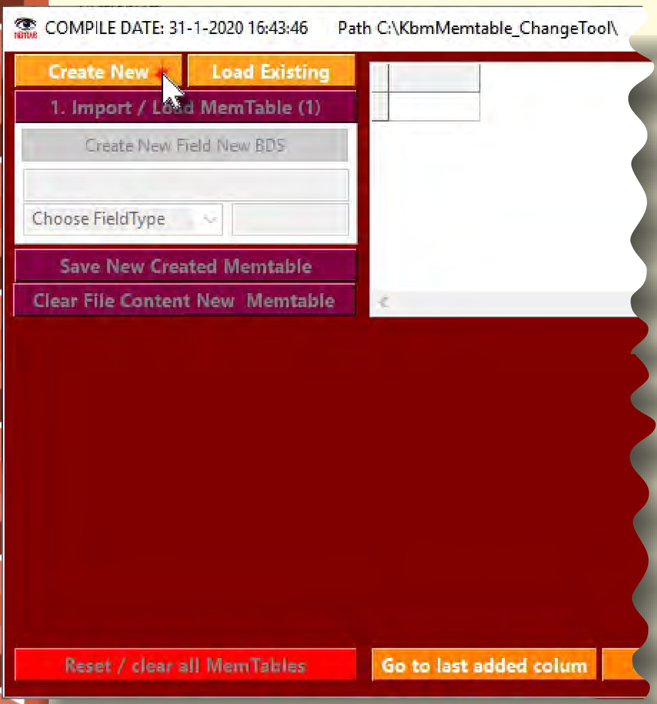

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_ChangeTool\
Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Choose FieldType
Save New Created Memtable
Clear File Content New Memtable
Reset / clear all MemTables
Go to last added colum

Figure 11: Create New starts the block of fields where you can enter the field values.

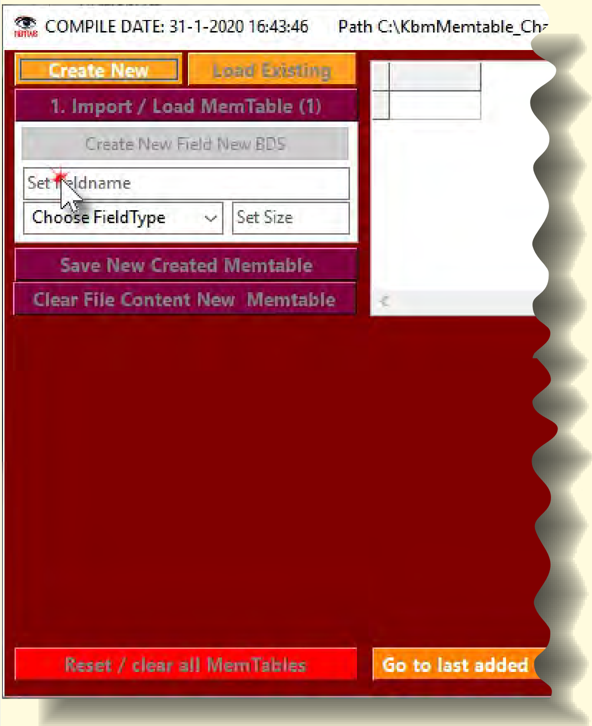
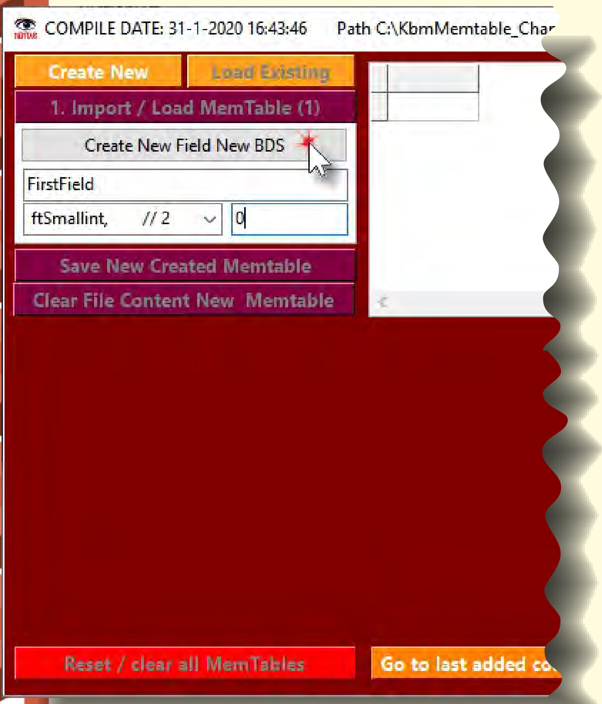

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_Ch
Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Set Fieldname
Choose FieldType
Set Size
Save New Created Memtable
Clear File Content New Memtable
Reset / clear all MemTables
Go to last added

Figure 12: Enter the values

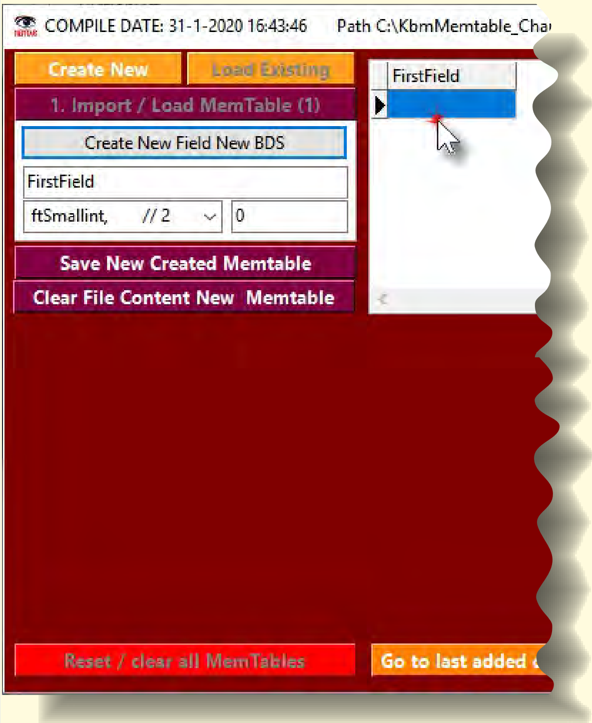

COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_Char

FirstField

ftSmallint, // 2

0
Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Save New Created Memtable
Clear File Content New Memtable
Reset / clear all MemTables
Go to last added co

Figure 13: After that you can create the new field


COMPILE DATE: 31-1-2020 16:43:46 Path C:\KbmMemtable_Cha

FirstField

ftSmallint, // 2

0

| FirstField |
|---|
| |
Create New
Load Existing
1. Import / Load MemTable (1)
Create New Field New BDS
Save New Created Memtable
Clear File Content New Memtable
Reset / clear all MemTables
Go to last added

Figure 14: After filling all the fields you can push the create field button and you see the first field in the grid. The Save and Clear button has become enabled.

C-QUEL

# In need of Delphi support ?

Because our people almost breathe Delphi you can ask us anything about Delphi. If you're in need of some extra hands to speed up your project, if you want to port your project to the web, if you want to slim your fat client, don't hesitate to give us a call.

## How can we help you ?

Contact us on +32 (11) 72 61 83 or mail us on info@cquel.be

Want to know more about us? See https://www.cquel.be

There is a Embarcadero Delphi MVP on the team and
we are tmssoftware.com certified technical partner for the Benelux.

s

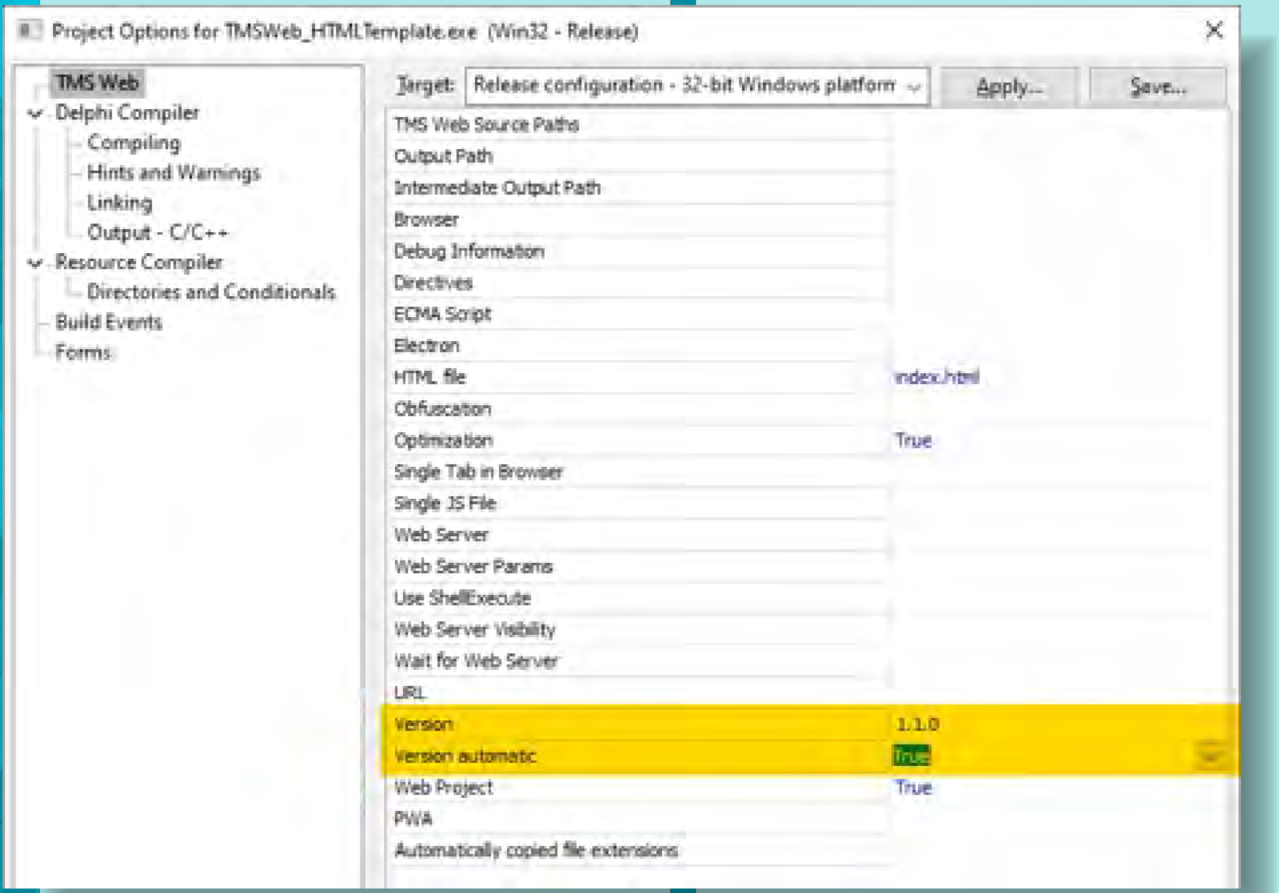
embarcadero MVP Delphi®

tmssoftware.com

https://www.cquel.be

Before, this was typically something that developers could manually perform after a compile is release mode was done. To save the effort of this extra step, a minify/uglify of the generated **JavaScript** code will automatically be performed after compiling. This typically **reduces the size** of the generated JavaScript code by 40%

## 2) Automatic JavaScript file versioning



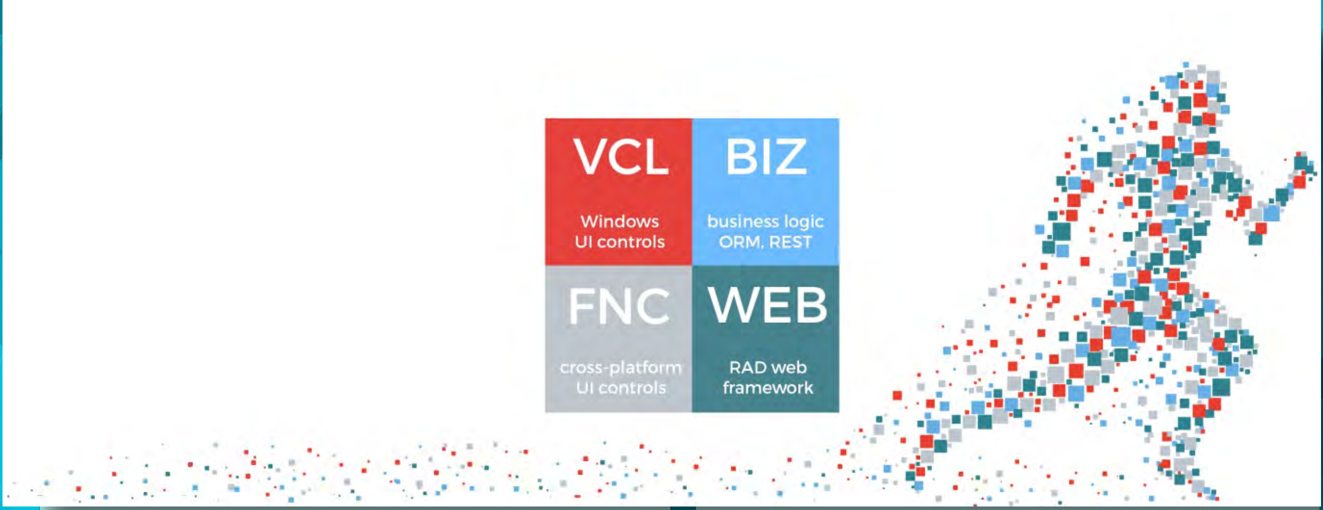
Project Options for TMSWeb_HTMLTemplate.exe (Win32 - Release)

- TMS Web
- Delphi Compiler
  - Compiling
  - Hints and Warnings
  - Linking
  - Output - C/C++
- Resource Compiler
  - Directories and Conditionals
- Build Events
- Forms

Target: Release configuration - 32-bit Windows platform

| | |
|---|---|
| TMS Web Source Paths | |
| Output Path | |
| Intermediate Output Path | |
| Browser | |
| Debug Information | |
| Directives | |
| ECMA Script | |
| Electron | |
| HTML file | index.html |
| Obfuscation | |
| Optimization | True |
| Single Tab in Browser | |
| Single JS File | |
| Web Server | |
| Web Server Params | |
| Use ShellExecute | |
| Web Server Visibility | |
| Wait for Web Server | |
| URL | |
| Version | 1.1.0 |
| Version automatic | True |
| Web Project | True |
| PWA | |
| Automatically copied file extensions | |

Now, upon each compile in release mode, the **JavaScript** filename can automatically have a version number suffix.
And this version number can be automatically incremented. **This brings the advantage that when a new version is released, there is no more risk that the browser keeps loading an old cached version of the project JavaScript file**.

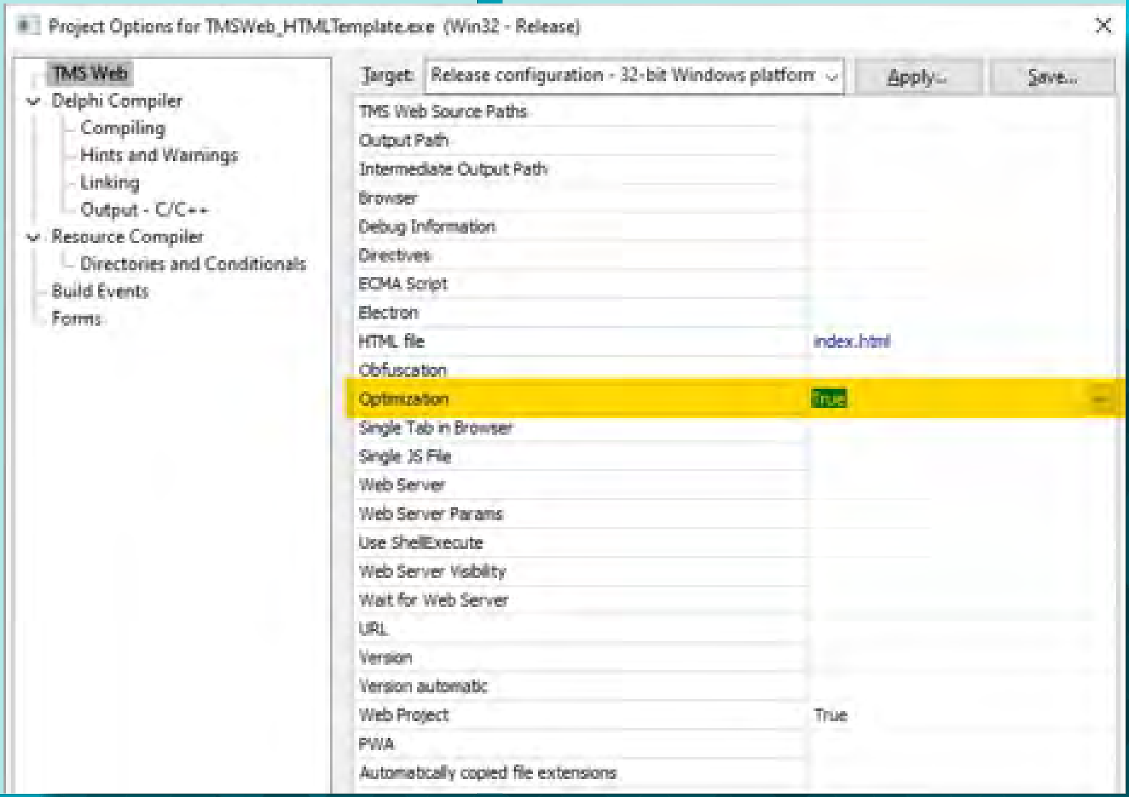


**TMS WEB Core v1.3 is the third major release of TMS WEB Core since its first release in 2018.**

With **TMS WEB Core**, a Pascal developer can use a RAD, component-based approach to create web client applications from the **Delphi IDE** or the **Lazarus IDE**.
The newest version v1.3 is a new milestone in our goal to make web client development for Pascal developers as convenient, fast and familiar as possible, encapsulating heaps of functionality in Pascal classes & components.

**In this article, we are going to have a look at the new v1.3 version via an overview of 20 new features.**
If you are not yet familiar with TMS WEB Core, please consult several extensive articles that have been written in prior **Blaise Pascal Magazine** issues on this topic. With this background, let's jump right into the list of new features & capabilities:

## 1) AUTOMATIC MINIFY/UGLIFY WHEN COMPILING IN RELEASE MODE FROM THE IDE

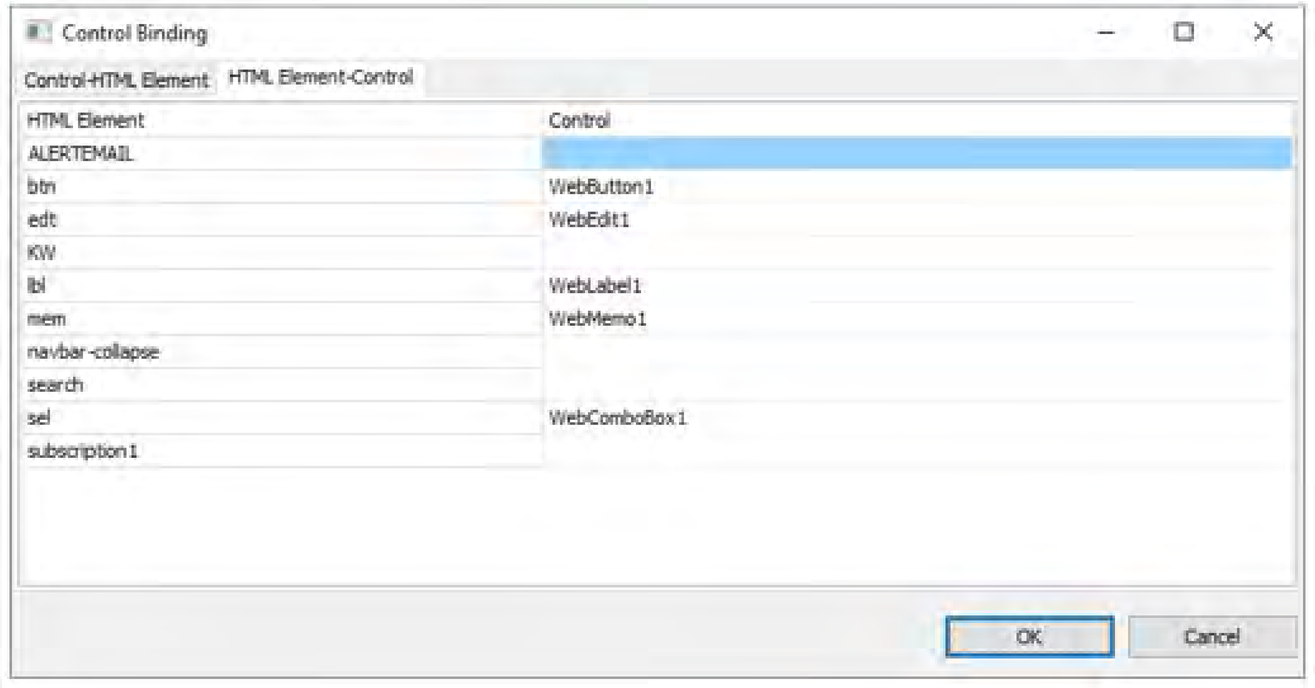

Project Options for TMSWeb_HTMLTemplate.exe (Win32 - Release)

- TMS Web
- Delphi Compiler
  - Compiling
  - Hints and Warnings
  - Linking
  - Output - C/C++
- Resource Compiler
  - Directories and Conditionals
- Build Events
- Forms

Target: Release configuration - 32-bit Windows platform

| | |
|---|---|
| TMS Web Source Paths | |
| Output Path | |
| Intermediate Output Path | |
| Browser | |
| Debug Information | |
| Directives | |
| ECMA Script | |
| Electron | |
| HTML file | index.html |
| Obfuscation | |
| Optimization | True |
| Single Tab in Browser | |
| Single JS File | |
| Web Server | |
| Web Server Params | |
| Use ShellExecute | |
| Web Server Visibility | |
| Wait for Web Server | |
| URL | |
| Version | |
| Version automatic | |
| Web Project | True |
| PWA | |
| Automatically copied file extensions | |

## 3) Easy UI for HTML element binding at design-time

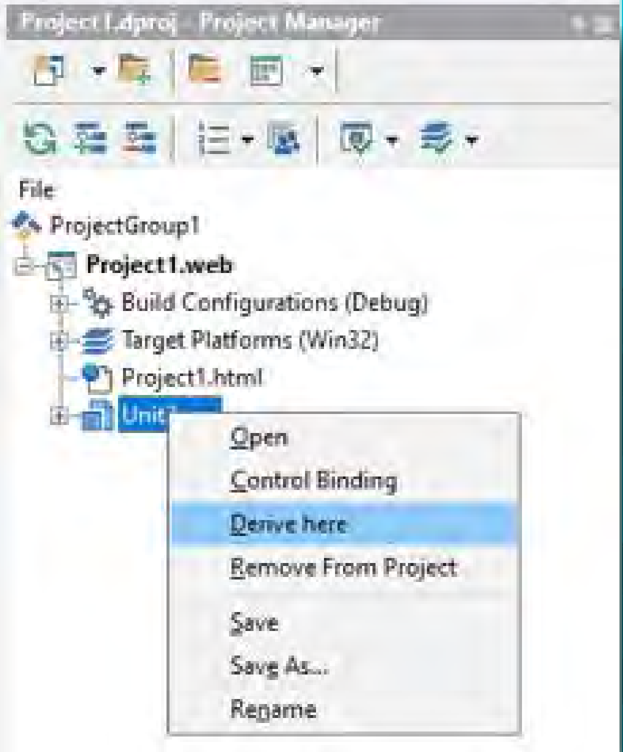


| HTML Element | Control |
|---|---|
| ALERTEMAIL | |
| btn | WebButton1 |
| edt | WebEdit1 |
| KW | |
| lbl | WebLabel1 |
| mem | WebMemo1 |
| navbar-collapse | |
| search | |
| sel | WebComboBox1 |
| subscription1 | |

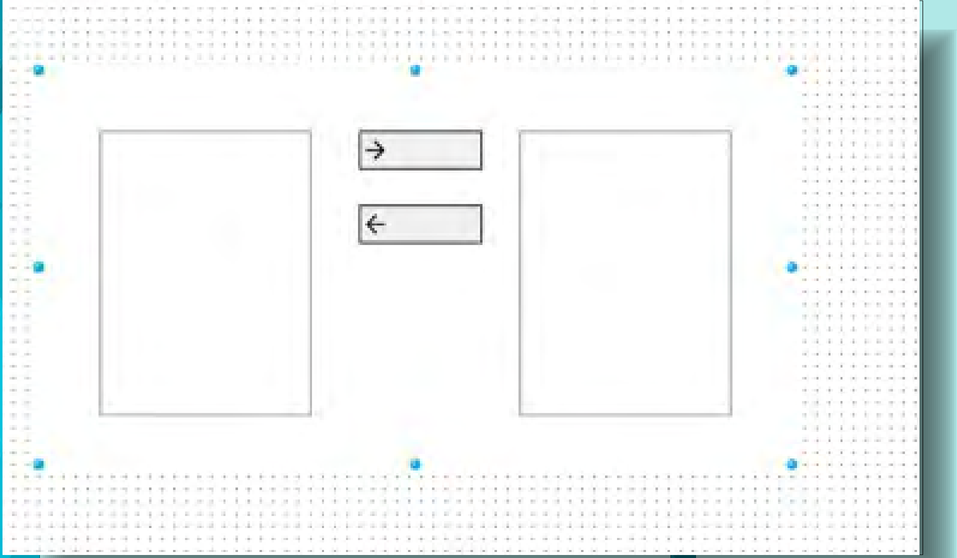
This is perhaps our favorite feature for **TMS WEB Core v1.3. By providing a grid view of all UI controls and the HTML elements in the template** these can be bound to or the opposite view of all HTML elements in the template and the UI controls there are linked to, it should help having a better overview of the HTML element template binding and to perform this task faster.

## 4) Visual form inheritance support

Yes, now you can have visual form inheritance from a **TMS WEB Core** web form. Simply right click the form from the project manager and select **"Derive here"**. This will add a new form to the project deriving from the selected form.

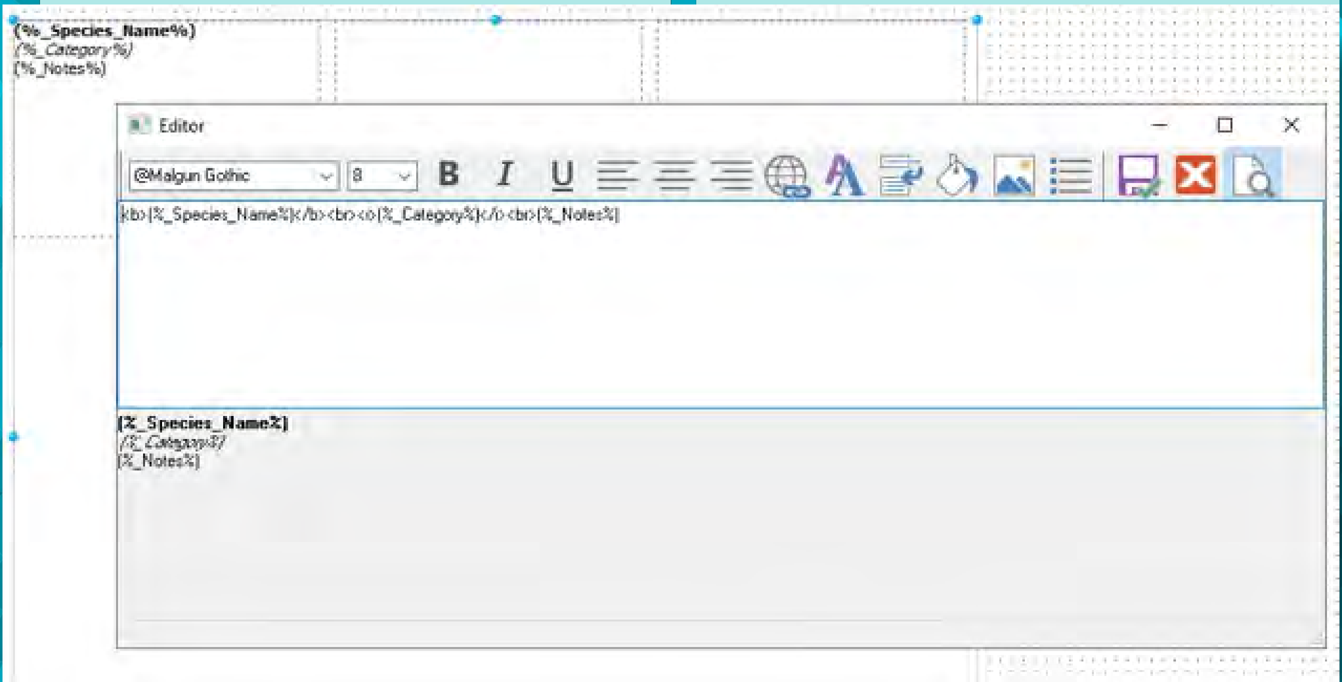


## 5) Frames support



The frames paradigm many Delphi developer like and use throughout applications is now available for **TMS WEB Core** projects as well.

## 6) Design-time editors for HTML template properties



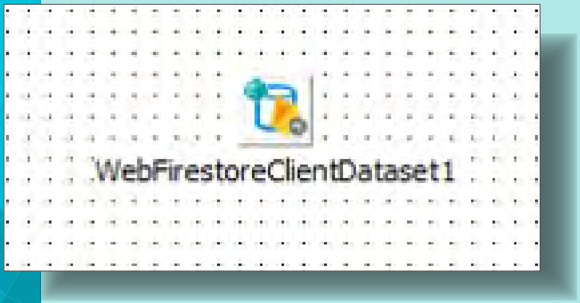
**(%_Species_Name%)**
*(%_Category%)*
(%_Notes%)

```
<b>(%_Species_Name%)</b><br><i>(%_Category%)</i><br>(%_Notes%)
```

**(%_Species_Name%)**
*(%_Category%)*
(%_Notes%)

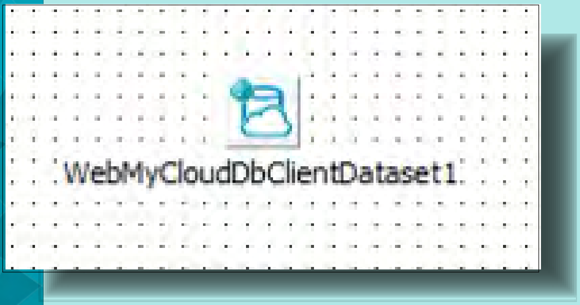
Some UI control properties allow for specifying HTML formatted text, for example as template for items in the **`TWebResponsiveGrid`.** Before, this HTML template was edited as a string in the Delphi IDE object inspector. As typically such HTML template string can be lengthy, this quickly became cumbersome to edit in the object inspector. Now, a popup editor appears that already offers some HTML formatting options from a toolbar and a preview of the HTML.

### 7) Firestore support via TWebFirestoreClientDataSet



If you want to take advantage of the **Google Firestore**
(See:
**`https://cloud.google.com/firestore/`**)
as a back-end for your data, this is now made even more simple with the **`TWebFirestoreClientDataSet`.**
This component provides access to a **Google Firestore** table as a dataset, offering full **CRUD** functionality. You can of course bind the **`TWebFirestoreClientDataSet`** to the various DB-aware TMS WEB Core UI controls via a **`TWebDataSource`.**

### 8) myCloudData.net support via TWebmyCloudDBClientDataset



The myCloudData.net (See: https://myclouddata.net) service offers cloud based data storage using tables with fixed fields and rich metadata. If you want to use the myCloudData.net service, the new component TWebmyCloudDBClientDataset makes it easier than ever to bind DB UI controls to its data.

### 9) SQLRestDBBridge support via TWebSQLRestClientDataset

The SQLRestDBBridge (See: **`https://wiki.freepascal.org/SQLDBRestBridge`**) is an open-source and free configurable **REST server to access SQL databases** on the server.
Also for this technology, a non-visual client dataset component **`TWebSQLRestClientDataset`** makes it easier than ever to use such server back-end with virtually no code to write.

## 10) Push notifications via TWebPushnotifications component

Website notification from site !
Your website push notification text would go here.
Google

Enable & use **push notifications** from your web application, also when the web application is not running. This technology uses the **Google Chrome, Firefox** or **Microsoft Edge web push notification server** to send notifications triggered from a back-end.
The `TWebPushnotifications` enables registering for such web push notifications.
We will also introduce later the back-end solution that controls the push notification generation server side.

## 11) Web crypto API wrapper classes

The web crypto API (See: `https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API`) is meanwhile also standard built-in in all modern browsers. Therefore, for your convenience, easy to use **Pascal wrapper classes** are now provided that allow you to take advantage from these APIs from your code.
This includes **AES & RSA** encryption as well as **RSA & HMAC** signature generation.

- `TAESEncryption`
- `TRSAEncryption`
- `TRSASignature`
- `THMACSignature`

### AES ENCRYPTION

*The Advanced Encryption Standard (AES), also known by its original name Rijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.*
`https://en.wikipedia.org/wiki/Advanced_Encryption_Standard`

### RSA ENCRYPTION

*SA (Rivest–Shamir–Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private). In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem".*
`https://en.wikipedia.org/wiki/RSA_(cryptosystem)`

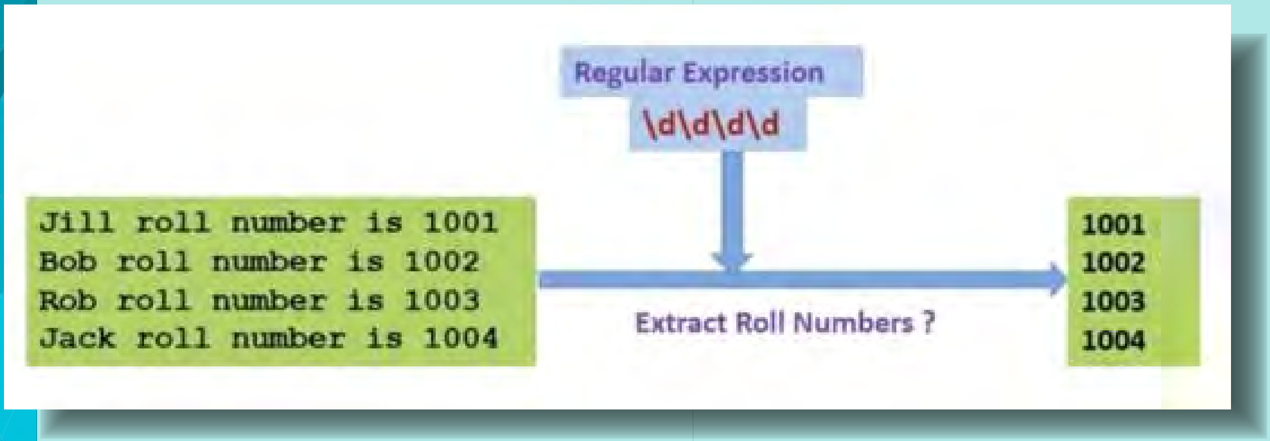
### RSASignature

*A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. A valid digital signature, where the prerequisites are satisfied, gives a recipient very strong reason to believe that the message was created by a known sender (authentication), and that the message was not altered in transit (integrity)*
`https://en.wikipedia.org/wiki/Digital_signature`

### HMAC Signature

*In cryptography, an HMAC (sometimes expanded as either keyed-hash message authentication code or hash-based message authentication code) is a specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authenticity of a message. Any cryptographic hash function, such as SHA-256 or SHA-3, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-X, where X is the hash function used (e.g. HMAC-SHA256 or HMAC-SHA3). The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and the size and quality of the key.*
`https://en.wikipedia.org/wiki/HMAC`

### 12) Regular expression API wrapper class TRegEx



Browsers did have regular expression handling functionality for a long time. To make this easy to use for Delphi developers, we created a `TRegEx class` with an almost identical interface to the Delphi `TRegEx` class for maximum code reusability.

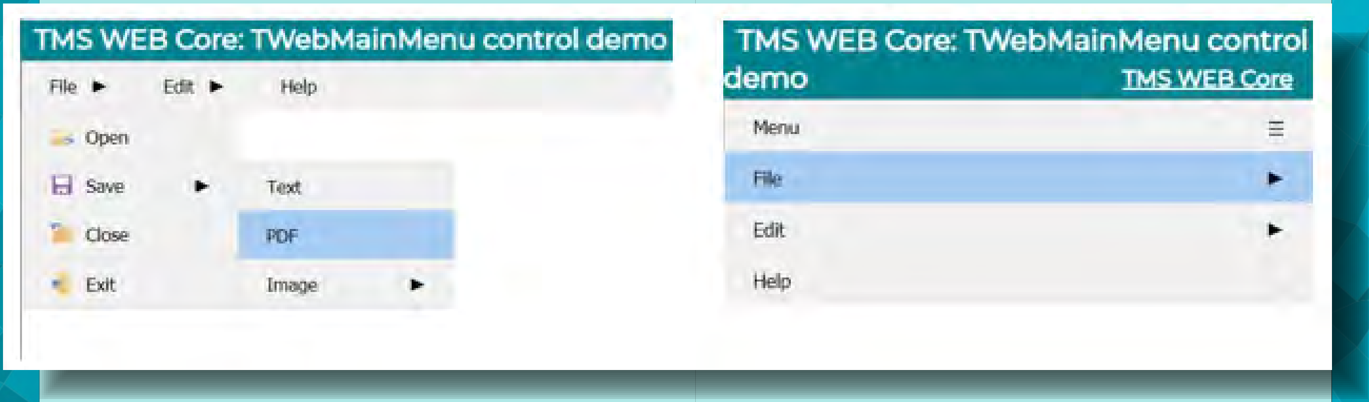
### 13) Local file access via TWebFilePicker, TWebFileUpload

`TWebFilePicker` is a component to let you pick local files via an open dialog while `TWebFileUpload` lets you do the same but in addition also enables to drag files from the machine file explorer to the browser. The components `TWebFilePicker, TWebFileUpload` have been extended to offer access to the properties of the selected local files as well as get the data of the file(s) in text, base64, URL or byte array format.

### 14) TWebLookupComboBox, TWebDBLookupComboBox

Having a combobox with two values per item, a displayed value and a data value is often a requirement for UI logic. With `TWebLookupComboBox` and its **DB-aware counterpart** `TWebDBLookupComboBox,` this is now out of the box available. Simply set & retrieve both the display value and data value per item.

### 15) Redesigned TWebMainMenu



Previously, the VCL `TMainMenu` equivalent for the web uses the **jqWidgets** controls by default. While the **jqWidgets** menu is nice & powerful, we thought it was better to have a menu component in the base controls and a separate one that is based on **jqWidgets** (See: `https://www.jqWidgets.com`) for those who need extra features.
The new `TWebMainMenu` however brings all the convenience of the VCL `TMainMenu` and in addition has built-in responsive design. That means, if the width of the menu becomes too wide, it will collapse automatically and behave as a hamburger menu.

## 16) New features for TWebGoogleMaps

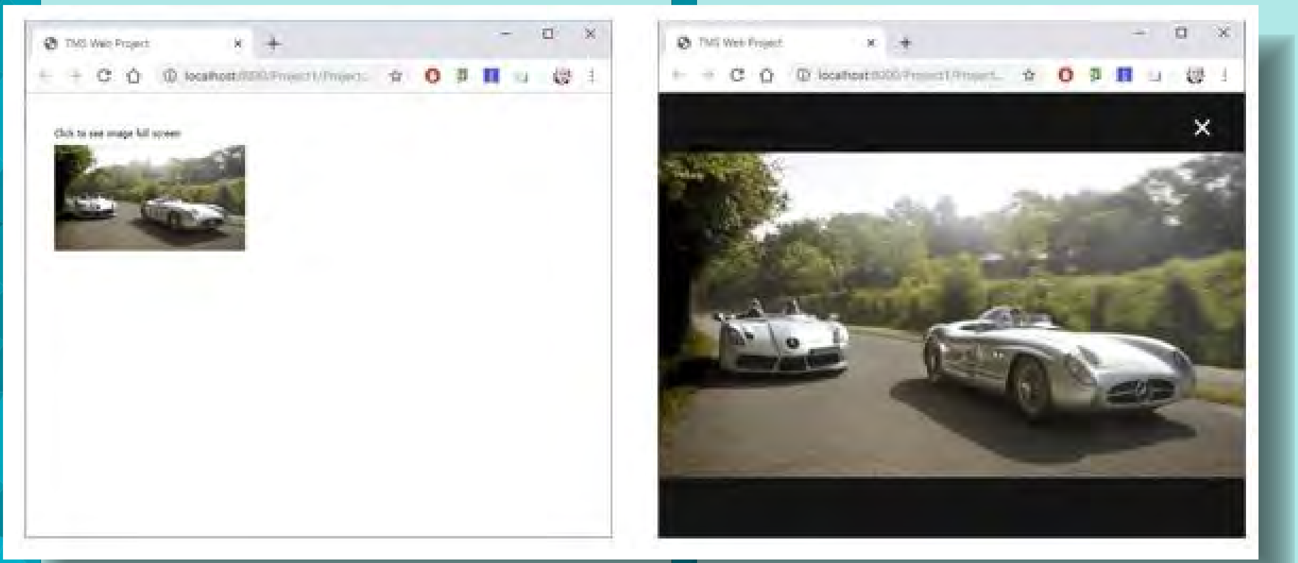
A lot of **Google Maps** functionality that was already available in our **VCL TMS WebGMaps** (See: **`https://www.tmssoftware.com/site/webgmaps.asp`**) or FMX equivalent for a long time has now been added to **`TWebGoogleMaps`**.
Adding circles, lines, rectangles, polygons, polylines is now possible.
Selecting themes, load **GPX** files, use **KML** layers is easily accessible by calling simple TWebGoogleMaps methods.

**GPX** Wikipedia
*A GPX file is a GPS data file saved in the GPS Exchange format, which is an open standard that can be freely used by GPS programs. It contains longitude and latitude location data that may include waypoints, routes, and tracks. GPX files are saved in XML format, which allows GPS data to be more easily imported and read by multiple programs and web services.*

**KML**
*Keyhole Markup Language (KML) is an XML notation for expressing geographic annotation and visualization within two-dimensional maps and three-dimensional Earth browsers. KML was developed for use with Google Earth, which was originally named Keyhole Earth Viewer.*

## 17) New TWebImageZoomControl

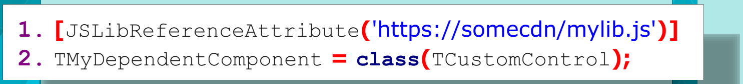
An often used paradigm in web applications is the **display of a thumbnail and showing the large version** of the image when it is clicked.
We have nicely encapsulated this functionality in the **`TWebImageZoomControl`**.
With this control, it is as easy as setting the thumbnail image URL and large image URL and the control does everything for you.


TMS Web Project
localhost:8000/Project1/Project...
Click to see image full screen
TMS Web Project
localhost:8000/Project1/Project...

### 18) Support for selecting the Electron version

With frequent new releases of the Electron framework (See: `https://www.electronjs.org/`) **for building cross platform desktop applications** from your web applications, controlling which version of Electron to use became hard. Now, with a setting under project options, it is easy to select with what version of Electron you want to package your application (if multiple Electron versions are installed on your machine).

### 19) Component attribute for JS & CSS lib dependencies



```
[JSLibReferenceAttribute('https://somecdn/mylib.js')]
TMyDependentComponent = class(TCustomControl);
```

This is a feature that will please developers diving into custom control development for **TMS WEB Core**. Often, a custom control can have a dependency to an external JavaScript and/or CSS library.

When one uses the control but forgets to include the references to these external libraries, the control will normally not work.

To avoid this, a new attribute was provided and the attribute parameters hold one or more external library references. When such control is added to the project, the library references will be automatically added to the project HTML file.

### 20) Update to latest pas2js compiler and pas2js RTL

We have included the latest pas2js compiler (See: `https://wiki.freepascal.org/pas2js`) release version and updated to the latest RTL source. This way you can take advantage of all latest releases of the open-source pas2js project.

## CONCLUSION

A colossal amount of work went into this new v1.3 release.

If you did not yet dive into this wonderful new world of web development capabilities for Delphi & Lazarus developers, grab a **fully functional trial download** for Delphi from `http://web.tmssoftware.com` or Blaise Pascal Magazine even offers a Lazarus install that comes with TMS WEB Core included at `https://www.blaisepascalmagazine.eu/9372-2/`

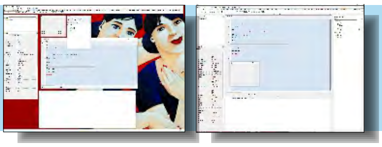
And we are not yet at the end of our road, work for v1.4 is already ongoing. Be involved, share your thoughts with the TMS software team and help steering the future of web development for Pascal developers!

# LAZARUS: DOCKING / UNDOCKING THE IDE PAGE 1/5

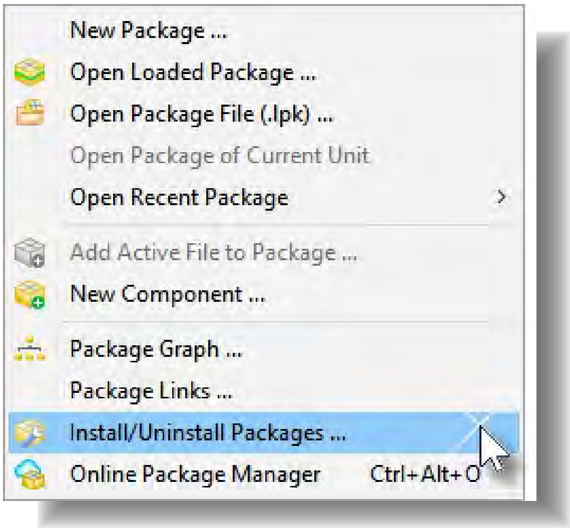
BY DETLEF OVERBEEK

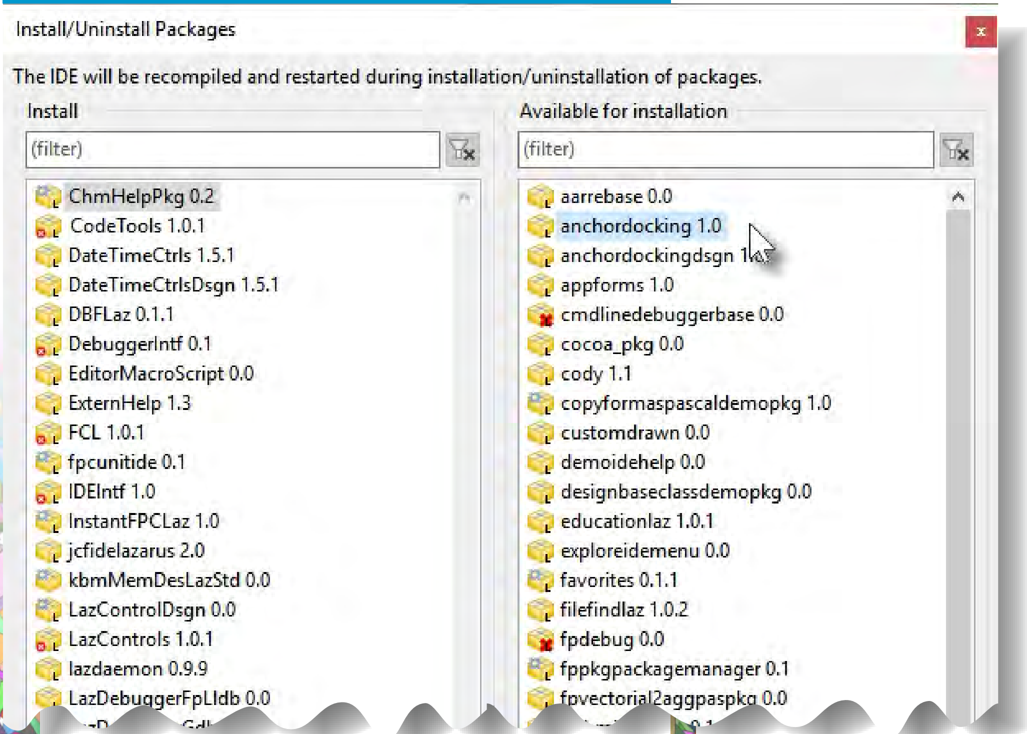
SHORT HOWTO

## INSTALLING



It is - as in Delphi - very simple to change from a standard undocked IDE (or vice versa) to a docked IDE which is the most comfortable environment for Laptops / Notebooks.
If you do not have several screens available or something like a 4k or even 8k screens, it is more workable. Lazarus has even as an **extra that the designform is still separately available on and of by the F12 key**.
All you need to do is install / uninstall it and then recompile. *(In Lazarus you can install components but after that you always need to recompile)*.
It has no designtime installed packages, which can be a very big advantage because **you practically never have to reinstall any of your installed components.** Even with a new release you simply do not have to do so.
(I'll explain HOWTO in an other article)
The installation of the docking is very easy: Click on the menu-item: (see right) Package -> Install / Uninstall Packages, the following window will show:


New Package ...
Open Loaded Package ...
Open Package File (.lpk) ...
Open Package of Current Unit
Open Recent Package
Add Active File to Package ...
New Component ...
Package Graph ...
Package Links ...
Install/Uninstall Packages ...
Online Package Manager Ctrl+Alt+O



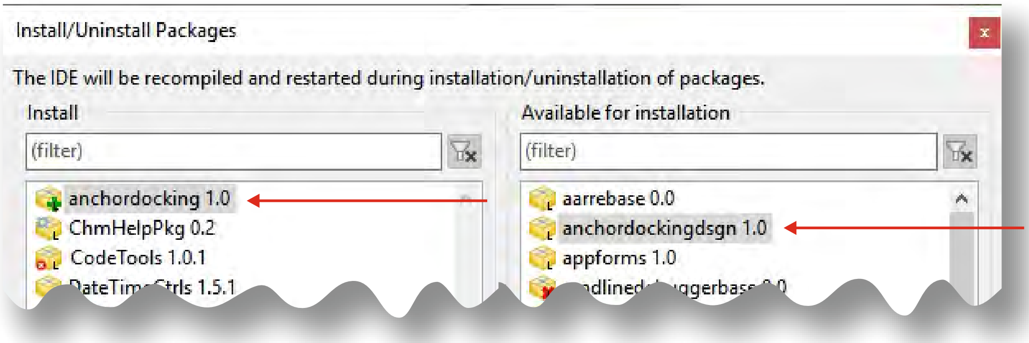
The IDE will be recompiled and restarted during installation/uninstallation of packages.

| Install | Available for installation |
|---|---|
| ChmHelpPkg 0.2 | aarrebase 0.0 |
| CodeTools 1.0.1 | anchordocking 1.0 |
| DateTimeCtrls 1.5.1 | anchordockingdsgn 1.0 |
| DateTimeCtrlsDsgn 1.5.1 | appforms 1.0 |
| DBFLaz 0.1.1 | cmdlinedebuggerbase 0.0 |
| DebuggerIntf 0.1 | cocoa_pkg 0.0 |
| EditorMacroScript 0.0 | cody 1.1 |
| ExternHelp 1.3 | copyformaspascaldemopkg 1.0 |
| FCL 1.0.1 | customdrawn 0.0 |
| fpcunitide 0.1 | demoidehelp 0.0 |
| IDEIntf 1.0 | designbaseclassdemopkg 0.0 |
| InstantFPCLaz 1.0 | educationlaz 1.0.1 |
| jcfidelazarus 2.0 | exploreidemenu 0.0 |
| kbmMemDesLazStd 0.0 | favorites 0.1.1 |
| LazControlDsgn 0.0 | filefindlaz 1.0.2 |
| LazControls 1.0.1 | fpdebug 0.0 |
| lazdaemon 0.9.9 | fppkgpackagemanager 0.1 |
| LazDebuggerFpLldb 0.0 | fpvectorial2aggpaspkg 0.0 |

Select anchor docking. You can not select more than one item at once. So if you want to install several items in one go, you need to select one and again an other: However, you can compile them in one go. It will of course lengthen the time you need to recompile.

In this case you need to install two items in the list: anchordocking and anchordockingdsgn.

You need to install them both because the packages do not work independently

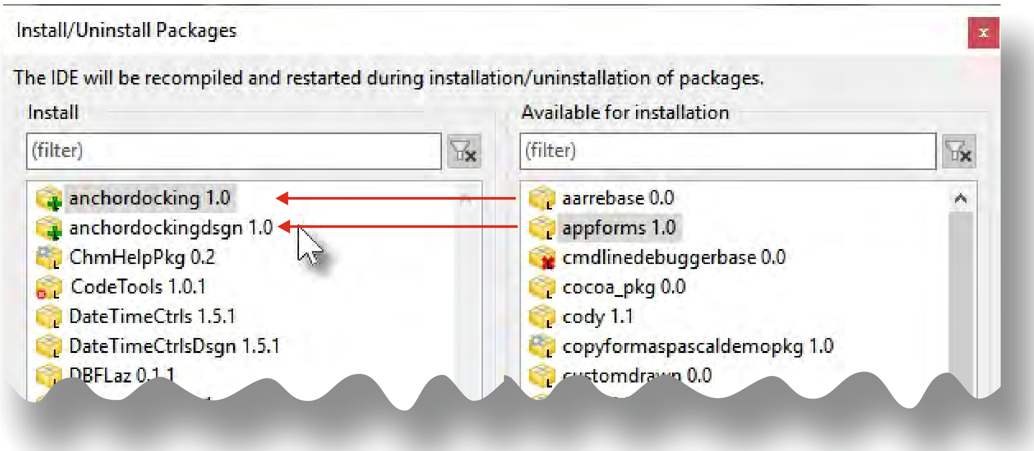


Install/Uninstall Packages

The IDE will be recompiled and restarted during installation/uninstallation of packages.

| Install | Available for installation |
|---|---|
| anchordocking 1.0 | aarrebase 0.0 |
| ChmHelpPkg 0.2 | anchordockingdsgn 1.0 |
| CodeTools 1.0.1 | appforms 1.0 |
| DateTimeCtrls 1.5.1 | cmdlinedebuggerbase 0.0 |

Now you can see the appearance of the  to be installed packages.

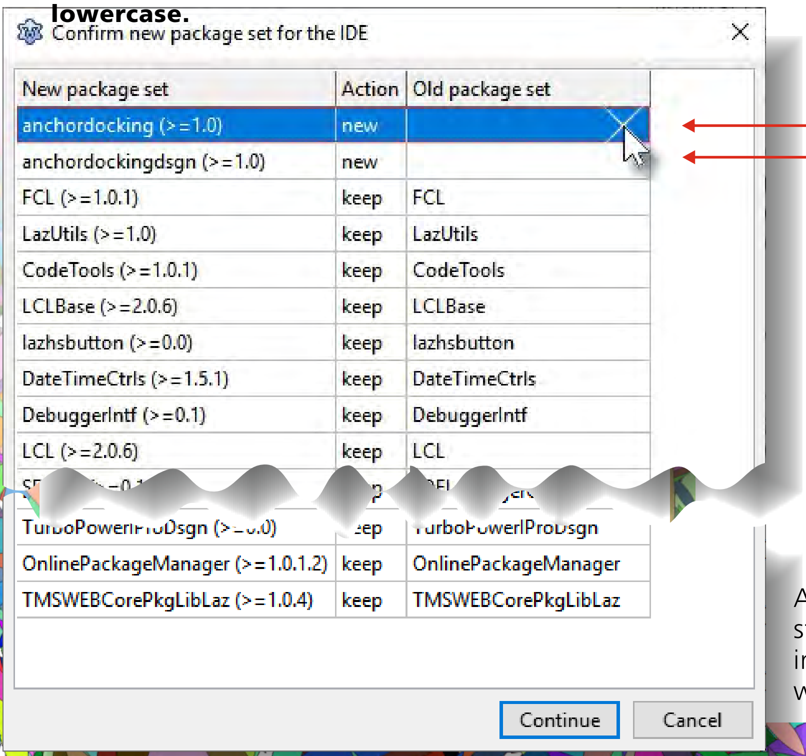


Install/Uninstall Packages

The IDE will be recompiled and restarted during installation/uninstallation of packages.

| Install | Available for installation |
|---|---|
| anchordocking 1.0 | aarrebase 0.0 |
| anchordockingdsgn 1.0 | appforms 1.0 |
| ChmHelpPkg 0.2 | cmdlinedebuggerbase 0.0 |
| CodeTools 1.0.1 | cocoa_pkg 0.0 |
| DateTimeCtrls 1.5.1 | cody 1.1 |
| DateTimeCtrlsDsgn 1.5.1 | copyformaspascaldemopkg 1.0 |
| DBFLaz 0.1.1 | customdrawn 0.0 |

You need to install as said before **both** components, **they depend on each other.** At the bottom of that window you can choose to **Save and Rebuild:** the next  window will appear which shows you the list containing your selected packages: they are the two at the top of the list: in the second column (Title: Action) you see that they are set to **new.** Notice that in this stadium the components are written in **lowercase.**



Confirm new package set for the IDE

| New package set | Action | Old package set |
|---|---|---|
| anchordocking (>=1.0) | new | |
| anchordockingdsgn (>=1.0) | new | |
| FCL (>=1.0.1) | keep | FCL |
| LazUtils (>=1.0) | keep | LazUtils |
| CodeTools (>=1.0.1) | keep | CodeTools |
| LCLBase (>=2.0.6) | keep | LCLBase |
| lazhsbutton (>=0.0) | keep | lazhsbutton |
| DateTimeCtrls (>=1.5.1) | keep | DateTimeCtrls |
| DebuggerIntf (>=0.1) | keep | DebuggerIntf |
| LCL (>=2.0.6) | keep | LCL |
| [illegible] | [illegible] | [illegible] |
| TurboPowerIProDsgn (>=0.0) | keep | TurboPowerIProDsgn |
| OnlinePackageManager (>=1.0.1.2) | keep | OnlinePackageManager |
| TMSWEBCorePkgLibLaz (>=1.0.4) | keep | TMSWEBCorePkgLibLaz |

After clicking **Continue** the compilation starts and then the packages will be installed: Lazarus will recompile and start up with the new docked IDE. (See next page)

# INTERNATIONAL PASCON EVENT 2020
# DELPHI & LAZARUS

BLAISE PASCAL MAGAZINE

DELPHI & LAZARUS
PASCON
EVENT

Tuesday 21 (Delphi) and Wednesday 22 (Lazarus) April 2020

DELPHI KEYNOTE & ROAD MAP / WEBCORE / RESTFRAMEWORK
FMXER / ANDREA MAGNI / KIM MADSEN / KBMMW ENTERPRISE
WEBASSEMBLY / PAS2JS & GENERICS / FPC 3.2 / LAZARUS 2.2
LAZARUS HANDBOOK PRESENTATION & SIGNATURE SESSION

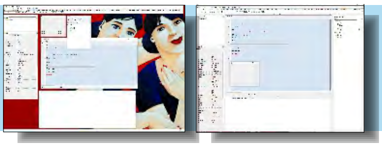
**The event address:**
"DE KLUIS" - Dr. Holtroplaan 1 Eindhoven / CONTACT: Mobile: +31 6 21.23.62.68
admin @ blaisepascalmagazine.eu / All English spoken. (German and Dutch translation available)

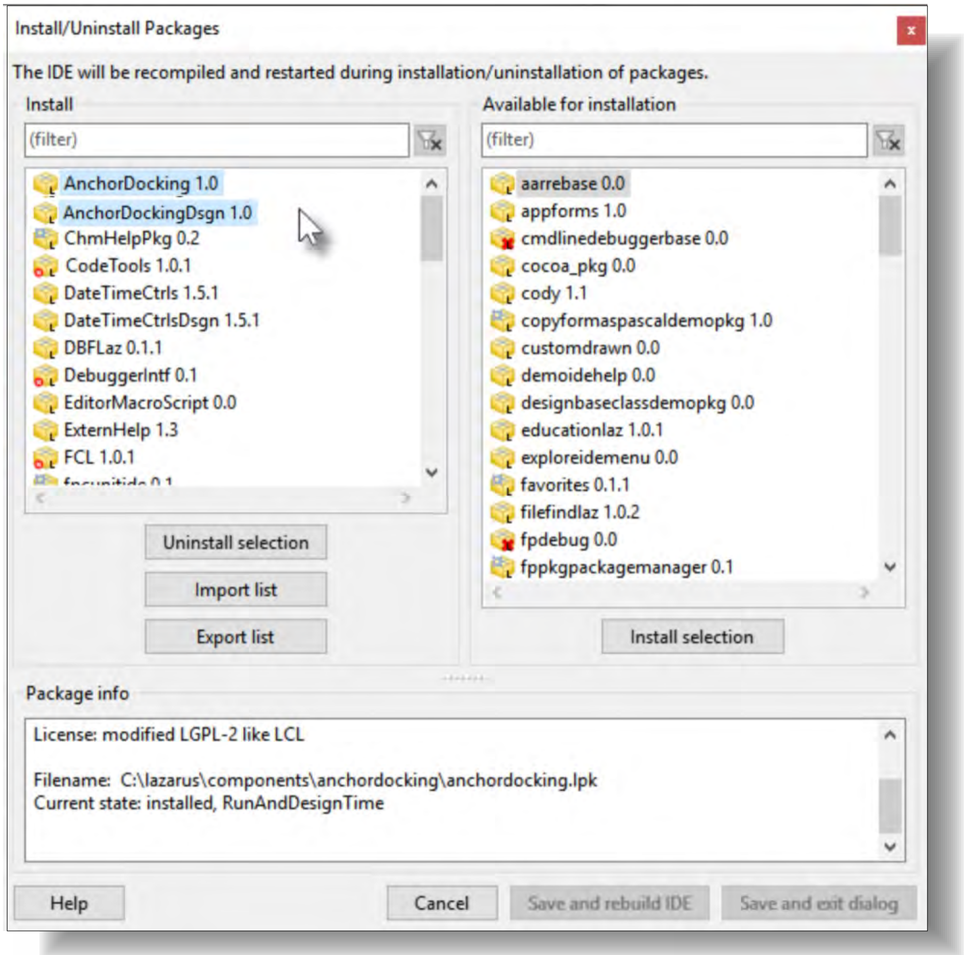
tmssoftware.com

barnsten

The Lazarus Factory

COMPONENTS 4 DEVELOPERS





Install/Uninstall Packages

The IDE will be recompiled and restarted during installation/uninstallation of packages.

Install

(filter)

- AnchorDocking 1.0
- AnchorDockingDsgn 1.0
- ChmHelpPkg 0.2
- CodeTools 1.0.1
- DateTimeCtrls 1.5.1
- DateTimeCtrlsDsgn 1.5.1
- DBFLaz 0.1.1
- DebuggerIntf 0.1
- EditorMacroScript 0.0
- ExternHelp 1.3
- FCL 1.0.1
- fpcunitide 0.1

Available for installation

(filter)

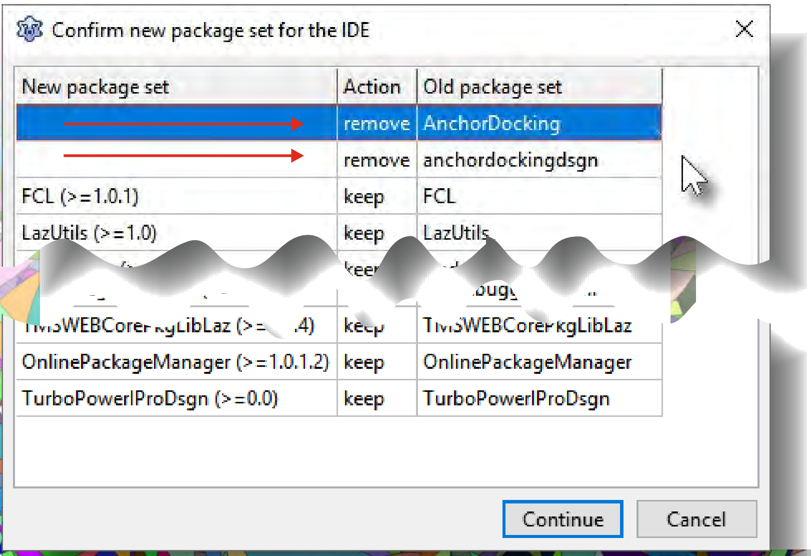
- aarrebase 0.0
- appforms 1.0
- cmdlinedebuggerbase 0.0
- cocoa_pkg 0.0
- cody 1.1
- copyformaspascaldemopkg 1.0
- customdrawn 0.0
- demoidehelp 0.0
- designbaseclassdemopkg 0.0
- educationlaz 1.0.1
- exploreidemenu 0.0
- favorites 0.1.1
- filefindlaz 1.0.2
- fpdebug 0.0
- fppkgpackagemanager 0.1

Package info

License: modified LGPL-2 like LCL

Filename: C:\lazarus\components\anchordocking\anchordocking.lpk
Current state: installed, RunAndDesignTime

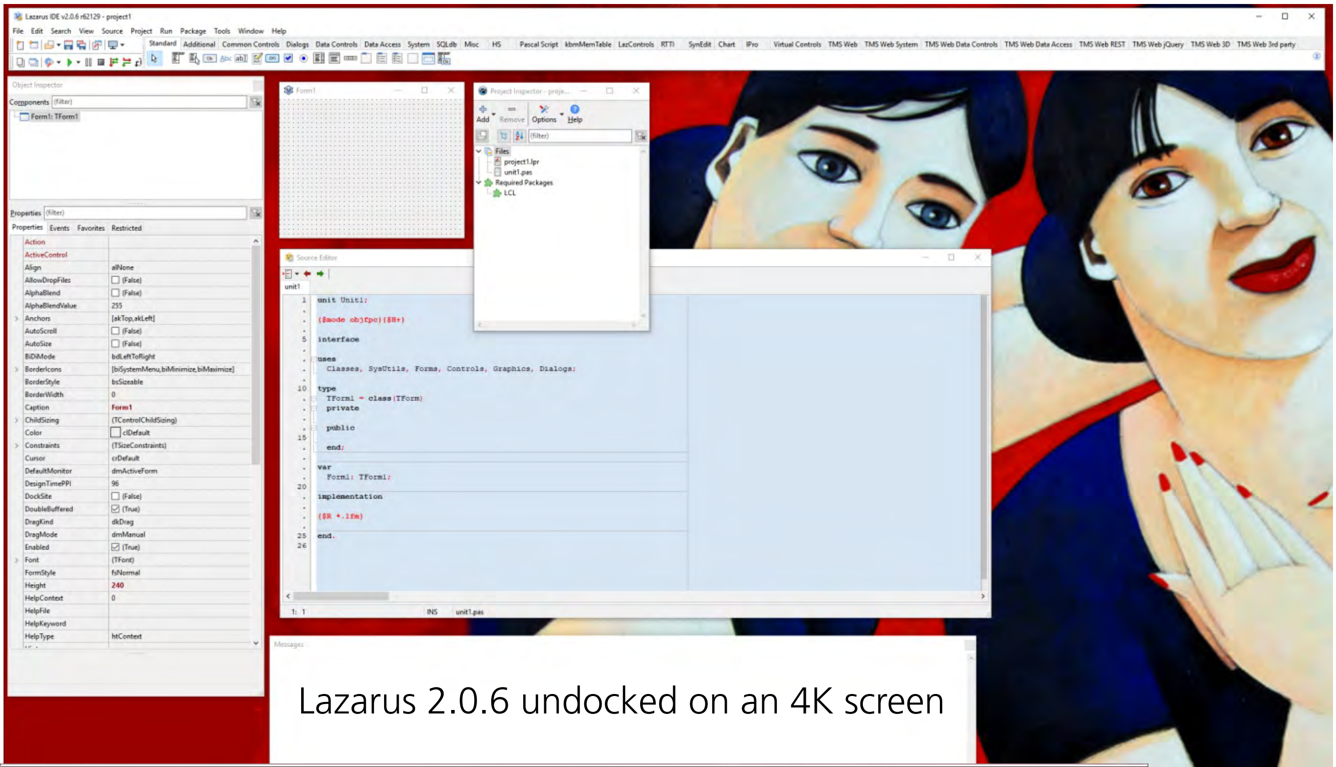
To uninstall: select in the left column **AnchorDocking** and **AnchordockingDsgn.** This time you can select more then one item. Notice that by now the name of the components is in **capitals.**



Confirm new package set for the IDE

| New package set | Action | Old package set |
|---|---|---|
| | remove | AnchorDocking |
| | remove | anchordockingdsgn |
| FCL (>=1.0.1) | keep | FCL |
| LazUtils (>=1.0) | keep | LazUtils |
| [illegible] | keep | [illegible] |
| [illegible] | [illegible] | [illegible]bug[illegible] |
| TMSWEBCorePkgLibLaz (>= .4) | keep | TMSWEBCorePkgLibLaz |
| OnlinePackageManager (>=1.0.1.2) | keep | OnlinePackageManager |
| TurboPowerIProDsgn (>=0.0) | keep | TurboPowerIProDsgn |

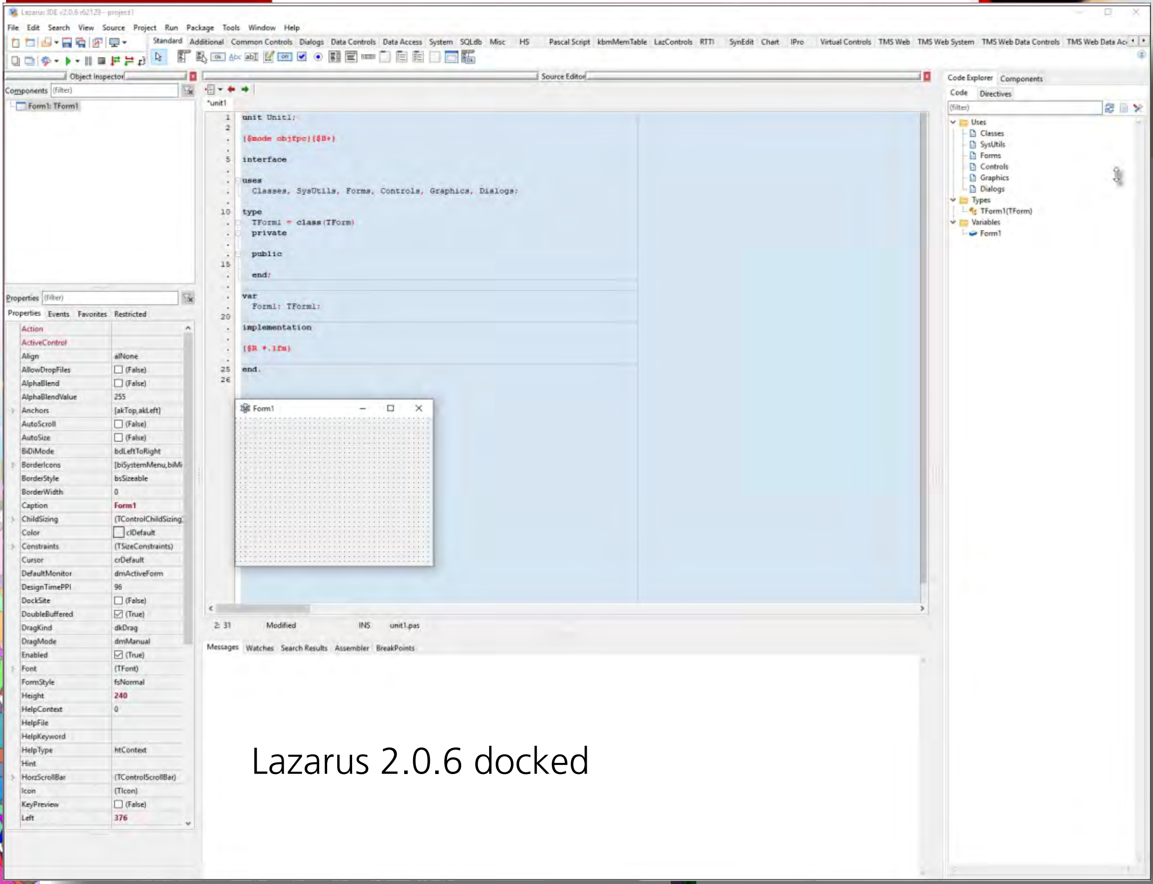
## THE RESULT


Lazarus IDE v2.0.6 r62129 - project1
File Edit Search View Source Project Run Package Tools Window Help
Standard Additional Common Controls Dialogs Data Controls Data Access System SQLdb Misc HS Pascal Script kbmMemTable LazControls RTTI SynEdit Chart IPro Virtual Controls TMS Web TMS Web System TMS Web Data Controls TMS Web Data Access TMS Web REST TMS Web jQuery TMS Web 3D TMS Web 3rd party
Object Inspector
Components (filter)
Form1: TForm1
Properties (filter)
Properties Events Favorites Restricted
Form1
Project Inspector - proje...
Add Remove Options Help
(filter)
Files
project1.lpr
unit1.pas
Required Packages
LCL
Source Editor
unit1
1: 1 INS unit1.pas
Messages

| | |
|---|---|
| Action | |
| ActiveControl | |
| Align | alNone |
| AllowDropFiles | □ (False) |
| AlphaBlend | □ (False) |
| AlphaBlendValue | 255 |
| Anchors | [akTop,akLeft] |
| AutoScroll | □ (False) |
| AutoSize | □ (False) |
| BiDiMode | bdLeftToRight |
| BorderIcons | [biSystemMenu,biMinimize,biMaximize] |
| BorderStyle | bsSizeable |
| BorderWidth | 0 |
| Caption | Form1 |
| ChildSizing | (TControlChildSizing) |
| Color | □ clDefault |
| Constraints | (TSizeConstraints) |
| Cursor | crDefault |
| DefaultMonitor | dmActiveForm |
| DesignTimePPI | 96 |
| DockSite | □ (False) |
| DoubleBuffered | ☑ (True) |
| DragKind | dkDrag |
| DragMode | dmManual |
| Enabled | ☑ (True) |
| Font | (TFont) |
| FormStyle | fsNormal |
| Height | 240 |
| HelpContext | 0 |
| HelpFile | |
| HelpKeyword | |
| HelpType | htContext |

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs;

type
  TForm1 = class(TForm)
  private

  public

  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

end.
```

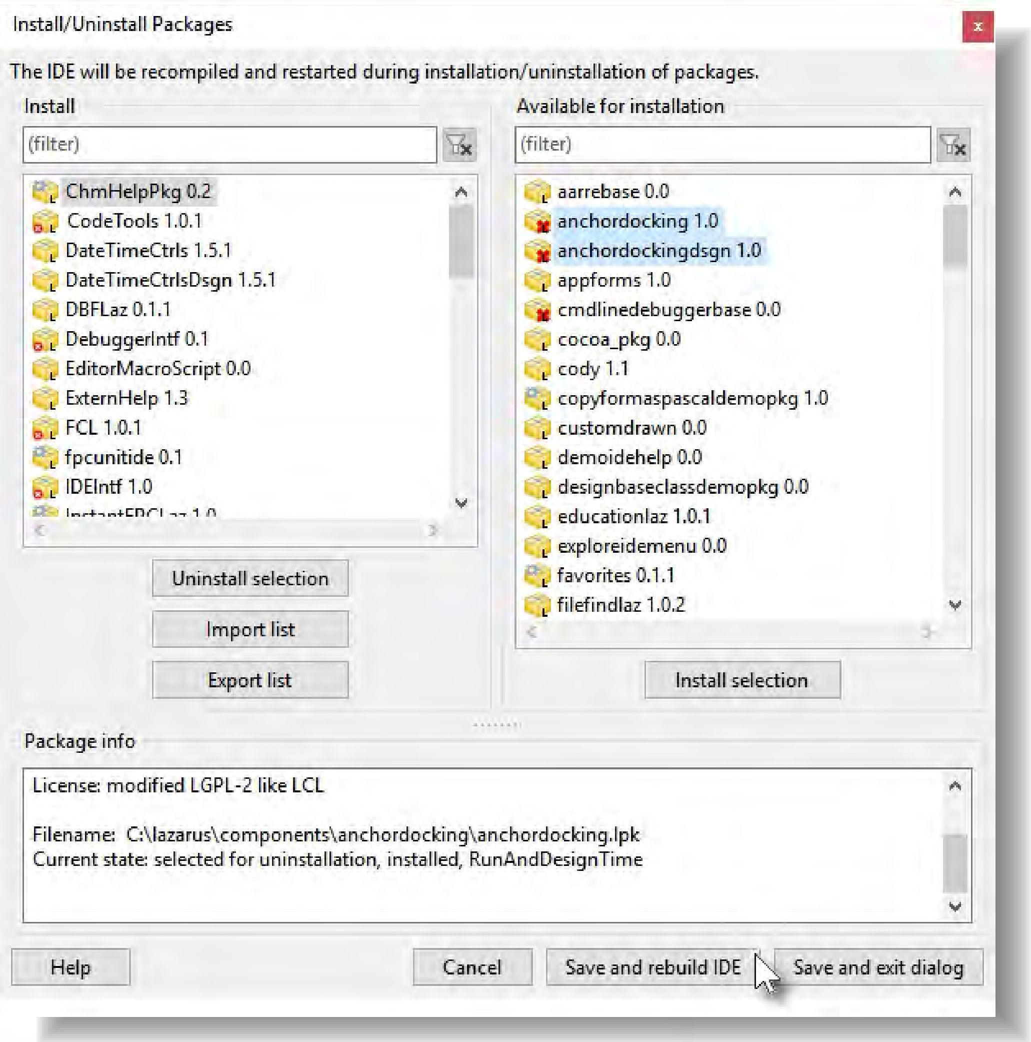
Lazarus 2.0.6 undocked on an 4K screen


Lazarus IDE v2.0.6 r62129 - project1
File Edit Search View Source Project Run Package Tools Window Help
Standard Additional Common Controls Dialogs Data Controls Data Access System SQLdb Misc HS Pascal Script kbmMemTable LazControls RTTI SynEdit Chart IPro Virtual Controls TMS Web TMS Web System TMS Web Data Controls TMS Web Data Ac
Object Inspector
Source Editor
Components (filter)
Form1: TForm1
Properties (filter)
Properties Events Favorites Restricted
*unit1
Form1
2: 31 Modified INS unit1.pas
Messages Watches Search Results Assembler BreakPoints
Code Explorer Components
Code Directives
(filter)
Uses
Classes
SysUtils
Forms
Controls
Graphics
Dialogs
Types
TForm1(TForm)
Variables
Form1

| | |
|---|---|
| Action | |
| ActiveControl | |
| Align | alNone |
| AllowDropFiles | □ (False) |
| AlphaBlend | □ (False) |
| AlphaBlendValue | 255 |
| Anchors | [akTop,akLeft] |
| AutoScroll | □ (False) |
| AutoSize | □ (False) |
| BiDiMode | bdLeftToRight |
| BorderIcons | [biSystemMenu,biMi |
| BorderStyle | bsSizeable |
| BorderWidth | 0 |
| Caption | Form1 |
| ChildSizing | (TControlChildSizing |
| Color | □ clDefault |
| Constraints | (TSizeConstraints) |
| Cursor | crDefault |
| DefaultMonitor | dmActiveForm |
| DesignTimePPI | 96 |
| DockSite | □ (False) |
| DoubleBuffered | ☑ (True) |
| DragKind | dkDrag |
| DragMode | dmManual |
| Enabled | ☑ (True) |
| Font | (TFont) |
| FormStyle | fsNormal |
| Height | 240 |
| HelpContext | 0 |
| HelpFile | |
| HelpKeyword | |
| HelpType | htContext |
| Hint | |
| HorzScrollBar | (TControlScrollBar) |
| Icon | (TIcon) |
| KeyPreview | □ (False) |
| Left | 376 |

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs;

type
  TForm1 = class(TForm)
  private

  public

  end;

var
  Form1: TForm1;

implementation

{$R *.lfm}

end.
```

Lazarus 2.0.6 docked



Install/Uninstall Packages

The IDE will be recompiled and restarted during installation/uninstallation of packages.

Install

(filter)

ChmHelpPkg 0.2
CodeTools 1.0.1
DateTimeCtrls 1.5.1
DateTimeCtrlsDsgn 1.5.1
DBFLaz 0.1.1
DebuggerIntf 0.1
EditorMacroScript 0.0
ExternHelp 1.3
FCL 1.0.1
fpcunitide 0.1
IDEIntf 1.0
InstantFPCLaz 1.0

Available for installation

(filter)

aarrebase 0.0
anchordocking 1.0
anchordockingdsgn 1.0
appforms 1.0
cmdlinedebuggerbase 0.0
cocoa_pkg 0.0
cody 1.1
copyformaspascaldemopkg 1.0
customdrawn 0.0
demoidehelp 0.0
designbaseclassdemopkg 0.0
educationlaz 1.0.1
exploreidemenu 0.0
favorites 0.1.1
filefindlaz 1.0.2

Package info

License: modified LGPL-2 like LCL

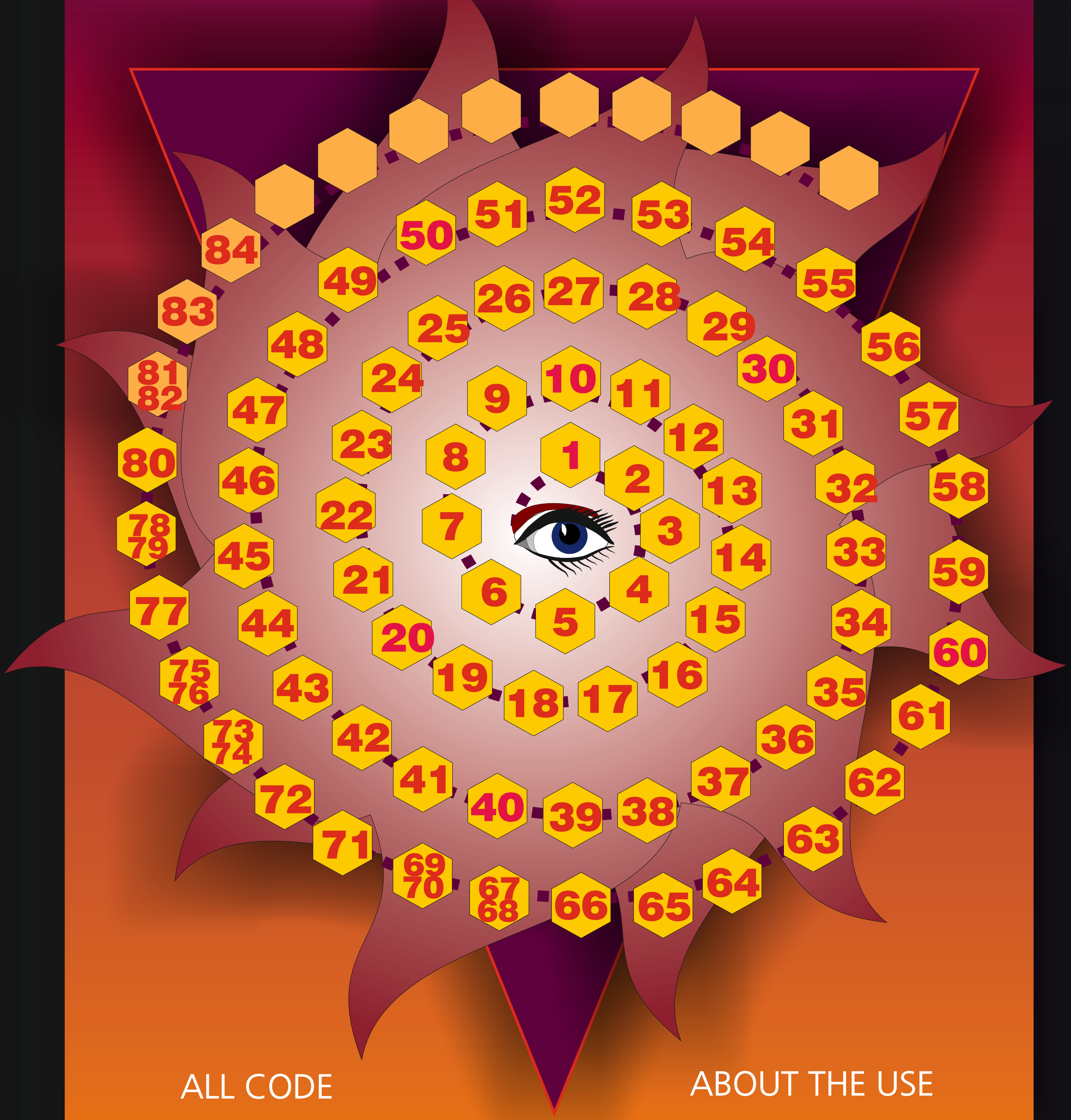
Filename: C:\lazarus\components\anchordocking\anchordocking.lpk
Current state: selected for uninstallation, installed, RunAndDesignTime

The result will be the both packages will be inserted to the right column. Notice that now the packages were inserted with **Capitals** and the right column with lower case and are marked with a red **X**.

After this just **save and rebuild** and you have your separated (undocked) IDE.

**That all there is to it.**

# LIBRARY 2020



ALL CODE

ABOUT THE USE

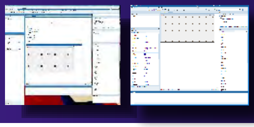
## BLAISE PASCAL MAGAZINE

ALL ISSUES IN ONE FILE

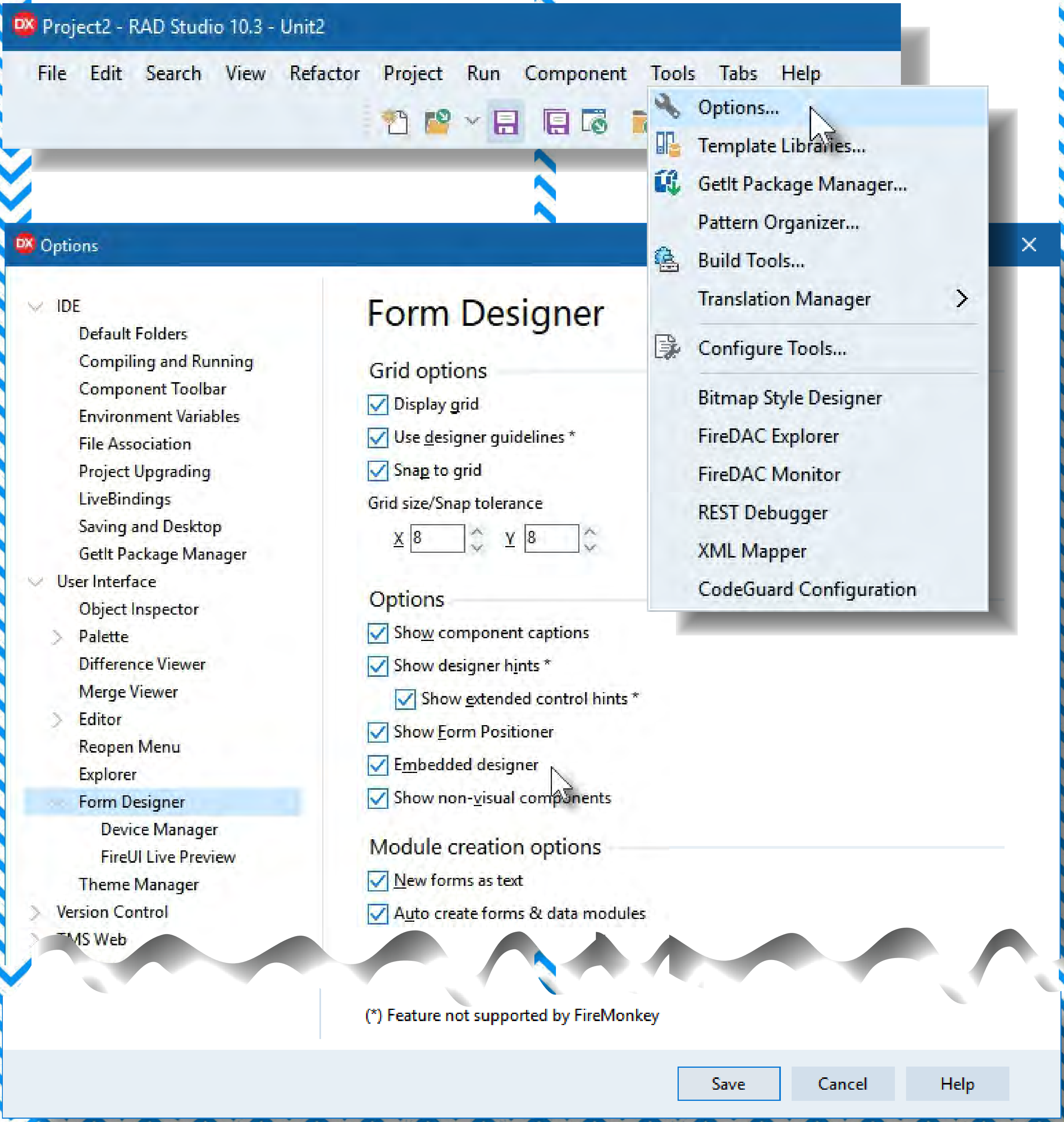
# DELPHI: DOCKING / UNDOCKING THE IDE PAGE 1/2

BY DETLEF OVERBEEK

SHORT HOWTO



It is in Delphi very simple to change from a standard docked IDE to an undocked IDE (or vice versa) which is the most comfortable environment for Laptops / Notebooks for Delphi as well as Lazarus. If you do not have several screens available or something like a 4k or even 8k screens, it is more workable. All you need to do is go to Tools -> Options -> User Interface -> Form Designer , submenu on the Form Designer choose Options to the selection Button 5 Embedded Designer and unmark the selection to which it is standard set. One thing still needs to be done: (after saving your work) restart Delphi. It will appear in undocked mode.
To reverse it you need to follow the same procedure and then mark the selection.



Form Designer

Grid options

☑ Display grid

☑ Use designer guidelines *

☑ Snap to grid

Grid size/Snap tolerance

X 8 Y 8

Options

☑ Show component captions

☑ Show designer hints *

☑ Show extended control hints *

☑ Show Form Positioner

☑ Embedded designer

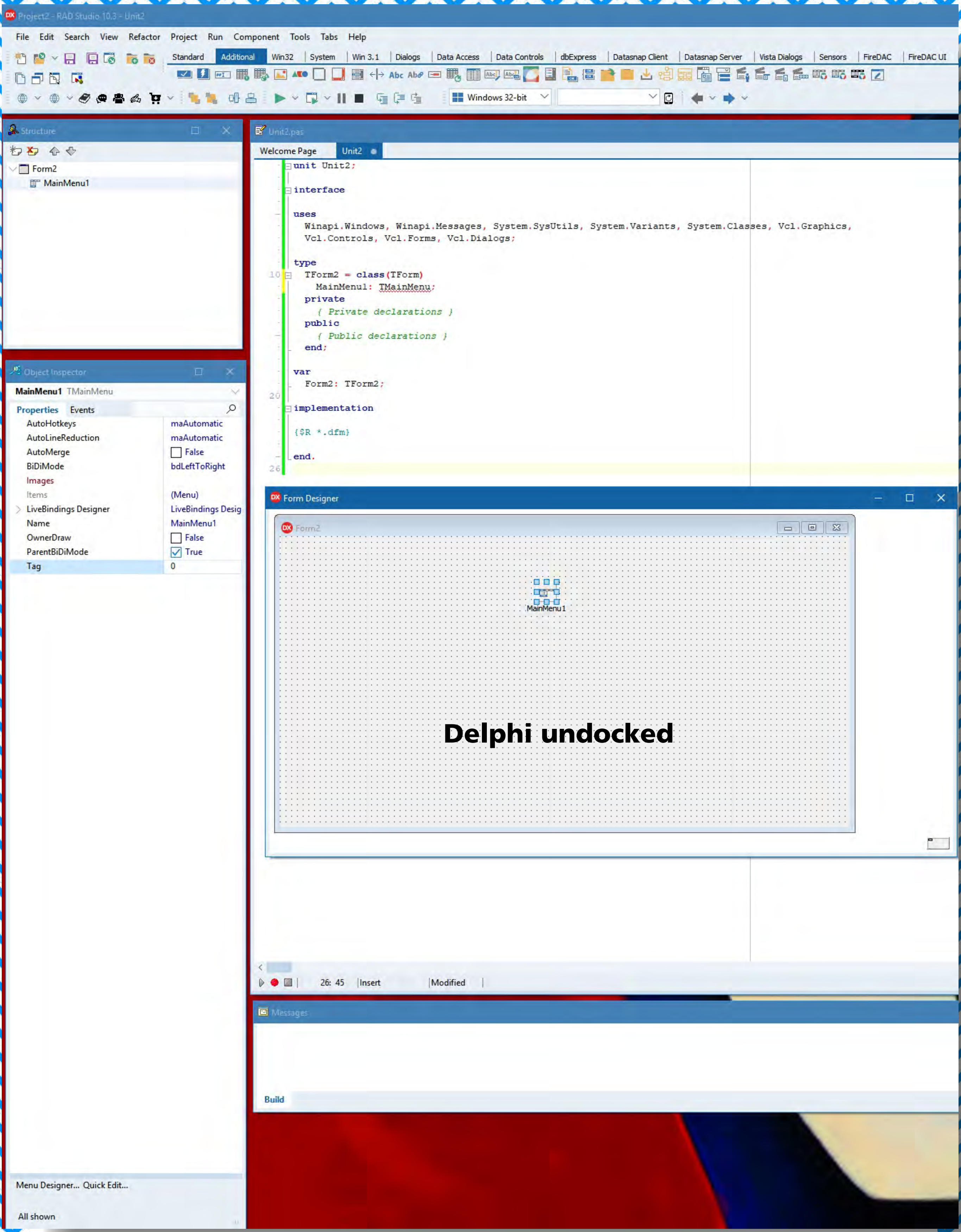
☑ Show non-visual components

Module creation options

☑ New forms as text

☑ Auto create forms & data modules

(*) Feature not supported by FireMonkey



```
unit Unit2;

interface

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs;

type
  TForm2 = class(TForm)
    MainMenu1: TMainMenu;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

end.
```

26: 45 | Insert | Modified

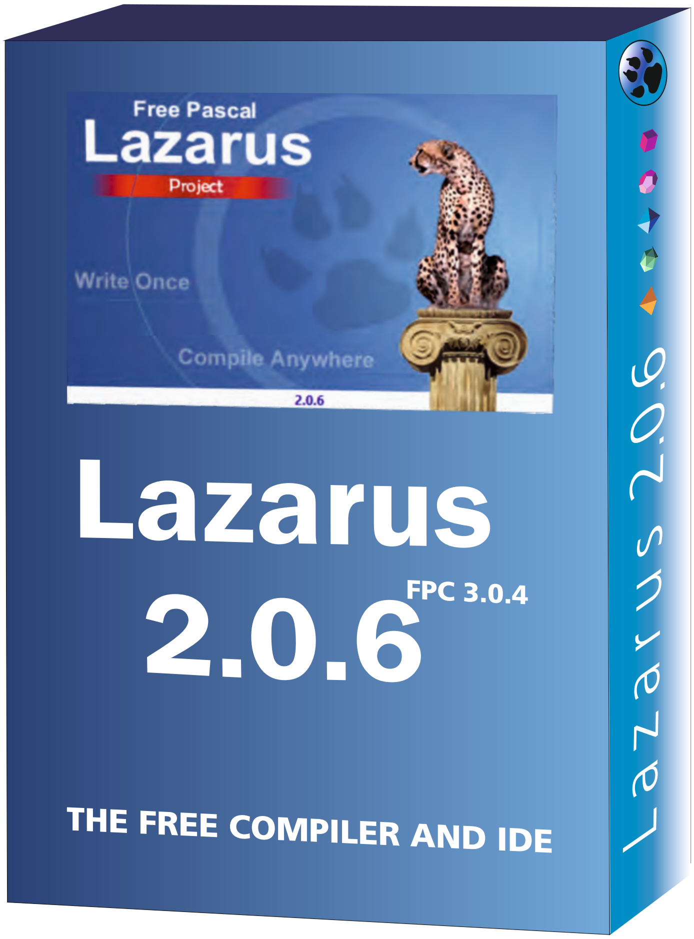
MainMenu1 TMainMenu

| | |
|---|---|
| AutoHotkeys | maAutomatic |
| AutoLineReduction | maAutomatic |
| AutoMerge | □ False |
| BiDiMode | bdLeftToRight |
| Images | |
| Items | (Menu) |
| LiveBindings Designer | LiveBindings Desig |
| Name | MainMenu1 |
| OwnerDraw | □ False |
| ParentBiDiMode | ☑ True |
| Tag | 0 |

**Delphi undocked**

That’s all there is to it.

The Lazarus Factory



**LAZARUS IS A DELPHI COMPATIBLE CROSS-PLATFORM IDE FOR FREE PASCAL.**
It includes LCL which is more or less compatible with Delphi's VCL. Free Pascal is a GPL'ed compiler that runs on Linux, Win32, OS/2, 68K RasberryPie and more. Free Pascal is designed to be able to understand and compile Delphi syntax, which is OOP. Lazarus is the part of the missing puzzle that will allow you to develop Delphi like programs in all of the above platforms.

**WHAT WIDGET SET?**
You decide. Lazarus is being developed to be totally and completely API independent. Once you write your code you just link it against the API widget set of your choice. If you want to use GTK+, great! If you want it to be Gnome compliant, great! As long as the interface code for the widget set you want to use is available you can link to it. If it isn't available, well you can write it.

**CAN YOU USE YOUR EXISTING DELPHI CODE?**
**IN GENERAL: YES.** If you are using some very specific databases, OCX, or DCU then the answer would be no. THESE ITEMS ARE SPECIFIC TO WINDOWS AND WOULD ONLY WORK ON AND WITHIN WINDOWS.

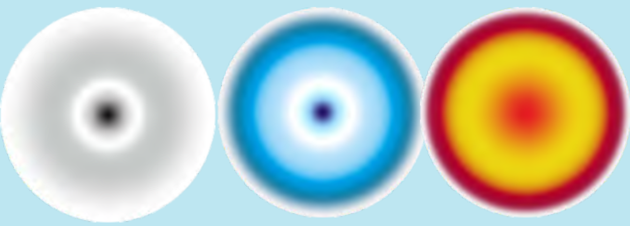
**CAN I CREATE COMMERCIAL PRODUCTS WITH THIS?**
**YES.** The code for the Free Pascal compiler is licensed under the GPL.

# COMPASSIST - THE ACTIVITY INDICATOR COMPONENT PAGE 1/5

starter expert

By Editor

COMPASSIST



In **Delphi Rio 10.3.3** there are some great items you should know. They will be shown here as projects so we can see how the component can be handled and what they are meant for. This is going to be a ongoing series and we will try to illuminate components that we think are extra interesting: **CompAssist**(Component Assistent)

**The first component I want to start with is the activity indicator, a very nice component that eases the end users worries: Why is nothing happening? - This is the tool that will show that there is activity going on. In itself it's fairly simple and can be used easily. We created a project that will show how things can be done. We have even added a number of extra images so you can make it more attractive. The project demonstrates the `TActivityIndicator` control and shows how to modify its various properties. We use the following controls:**

**THE ACTIVITY INDICATOR. (AI)**

`grpIndicatorSize`:
Sets the size of the activity indicator.

`grpIndicatorColor`:
Sets the color of the activity indicator.

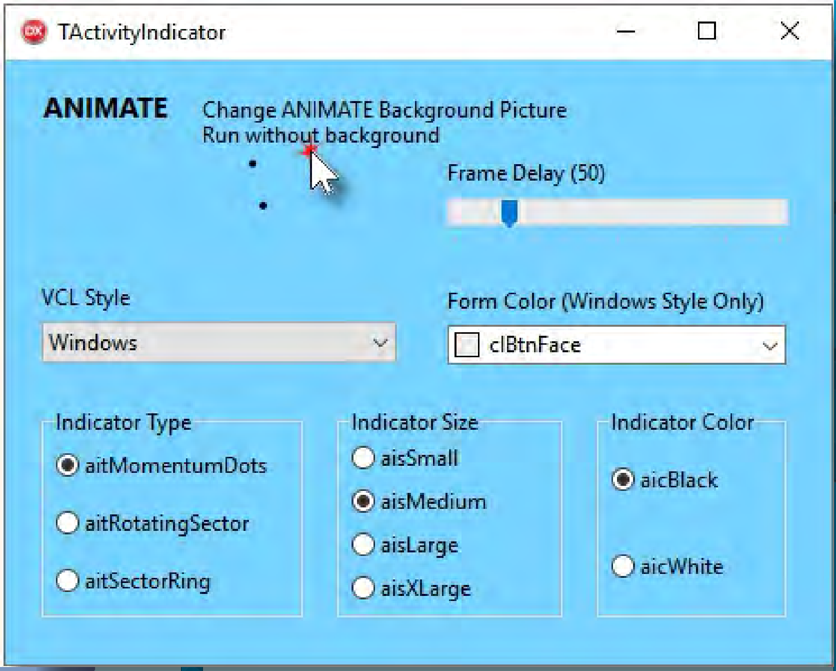
`grpIndicatorType`:
Sets the type of the activity indicator.

`trkFrameDelay`:
Sets the frame delay of the activity indicator.

`chkAnimate`:
Toggles the Animate property of the activity indicator.

`cbxFormColor`:
Sets the Color of the form if the current style is Windows.

`cbxVclStyles`:
A combo box that allows you to change the style of the application. You can choose between any style that is active in the Application Appearance - Custom Styles options for this project.



TActivityIndicator

**ANIMATE** Change ANIMATE Background Picture
Run without background

Frame Delay (50)

VCL Style: Windows

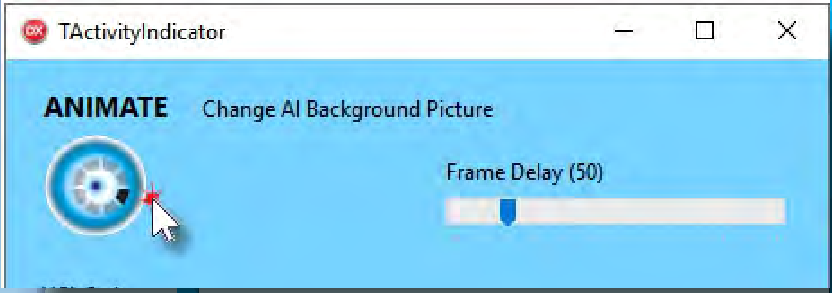
Form Color (Windows Style Only): □ clBtnFace

Indicator Type
- ● aitMomentumDots
- ○ aitRotatingSector
- ○ aitSectorRing

Indicator Size
- ○ aisSmall
- ● aisMedium
- ○ aisLarge
- ○ aisXLarge

Indicator Color
- ● aicBlack
- ○ aicWhite

**Figure 1: Overview**

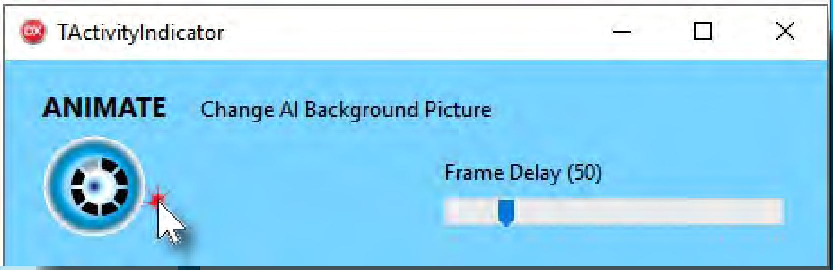


TActivityIndicator

**ANIMATE** Change AI Background Picture

Frame Delay (50)

**Figure 2: Animation running with background**



TActivityIndicator

**ANIMATE** Change AI Background Picture

Frame Delay (50)

**Figure 3: An other animation showing**

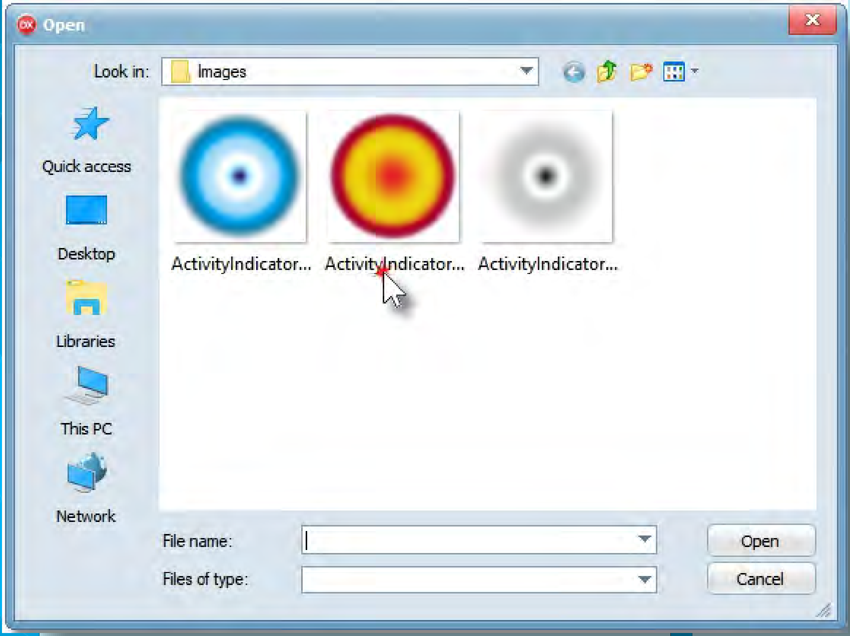


**Figure 4: Selection window for the background image**

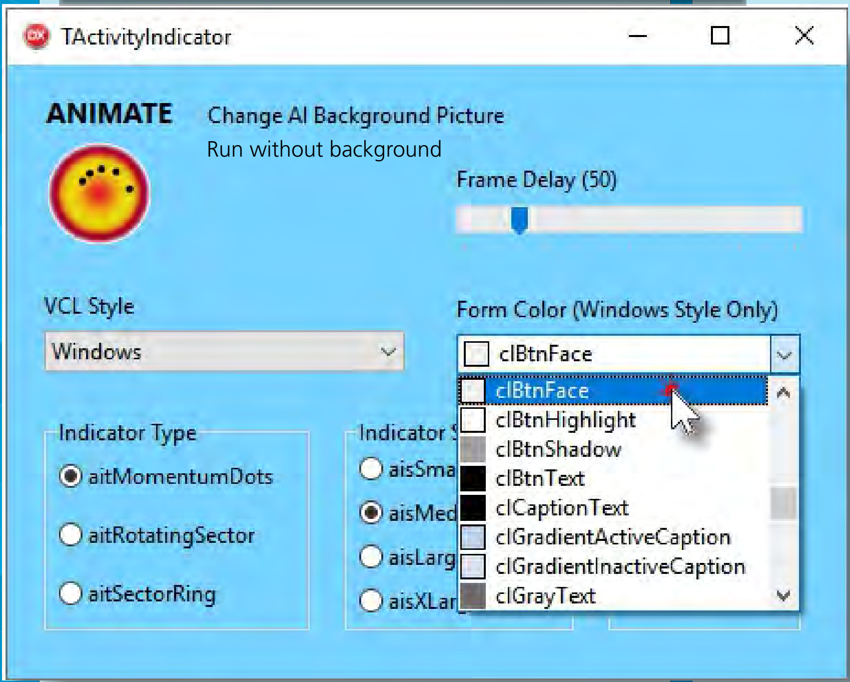
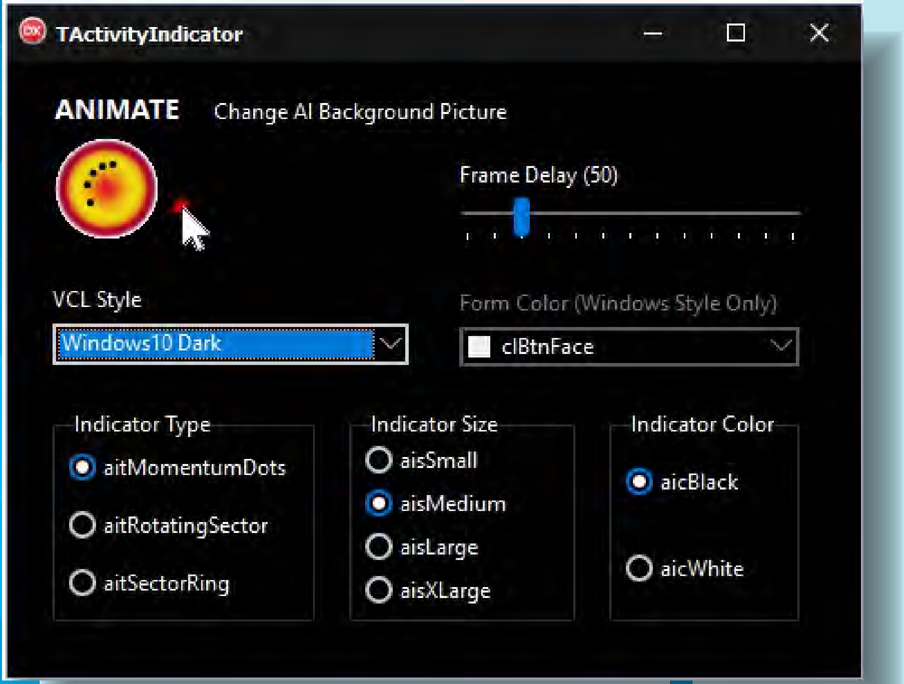


**Figure 5: Selecting windows style**

`TImage` Picture with various images that might be used to set the background area and change its location so it will be on the exact place you need it whenever you change the size of the activity indicator.
We have chosen to add a few features that make it easier to demonstrate the project:
Clicking on the
various Labels like ANIMATE will provoke action:

1: The label **Run without background** shows the action as is.
2: The label **Change ANIMATE Background Picture** will pop up a selection window where you can choose from.
3: Clicking **ANIMATE** will toggle the Activity Indicator with background the background can be altered or renewed.



TActivityIndicator

ANIMATE Change AI Background Picture

Frame Delay (50)

VCL Style: Windows10 Dark

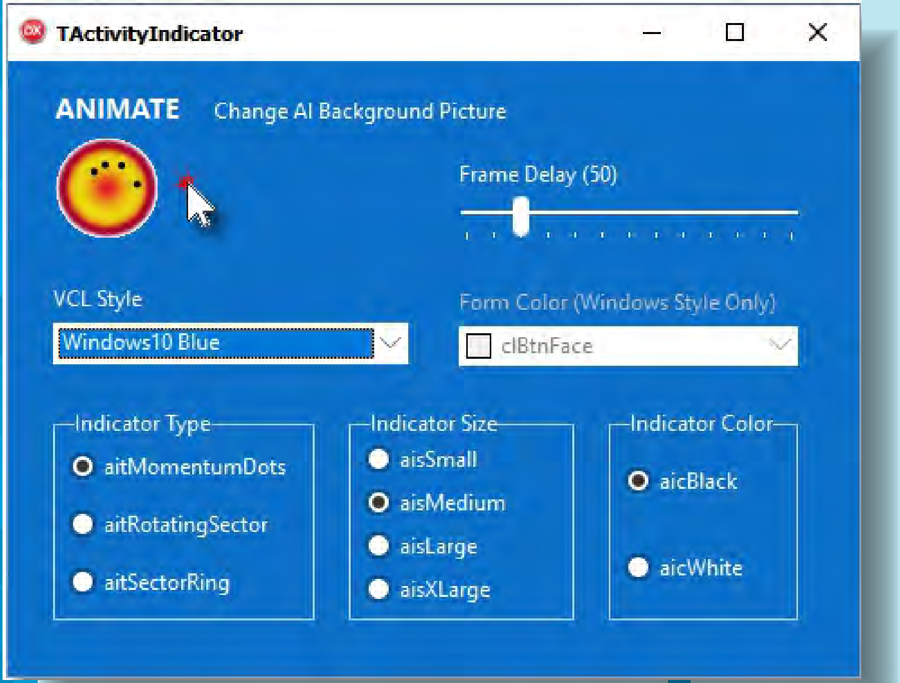
Form Color (Windows Style Only): clBtnFace

Indicator Type
● aitMomentumDots
○ aitRotatingSector
○ aitSectorRing

Indicator Size
○ aisSmall
● aisMedium
○ aisLarge
○ aisXLarge

Indicator Color
● aicBlack
○ aicWhite

**Figure 6: Selecting VCL Style**



TActivityIndicator

ANIMATE Change AI Background Picture

Frame Delay (50)

VCL Style: Windows10 Blue

Form Color (Windows Style Only): clBtnFace

Indicator Type
● aitMomentumDots
○ aitRotatingSector
○ aitSectorRing

Indicator Size
○ aisSmall
● aisMedium
○ aisLarge
○ aisXLarge

Indicator Color
● aicBlack
○ aicWhite

**Figure 7: An other color set by VCL style**

The Lazarus Factory

# LAZARUS SPECIAL EDITION 2.0.6

THIS PROGRAM IS FREE (YOU CAN DOWNLOAD IT HERE:
**(https://www.blaisepascalmagazine.eu/9372-2/)**
or from our website
**Colourbuttons,(HS)** Free including Code,
**Webcore (TMS)** Free, fully functional, no code
**kbmMemtable**(Standard Version)
**Components4DevelopersFree,** fully functional, no code

You can unpack this zip file and simply copy it to any directory, even a USB stick and it will work.

**You can compile & add other components to this version.**
**Do NOT do:** CleanUp and Build from the Lazarus Menu.
Then you will damage the files for this version, because it can NOT recompile the sources.
But you probably will never have to...
Be sure to have a copy of this on your system

## IMPORTANT:

Request a trial license:
**https://www.tmssoftware.com/site/trialkey.asp**
You need to install the TMS Webcore Trial
**TMSWEBCoreXE12_BIN.zip**
You can install only Webcore for Lazarus or install it as well for Delphi.

## Lazarus TMSDemo Projecten

**(https://www.blaisepascalmagazine.eu/wp-content/uploads/2019/12/LazarusTMSDemoProjecten.zip)** or go to our website

Align \ Anchors\ Bootstrap \ DataModule \ Dataset \ DBGrid \ EditAutoComplete \ FilePicker \ Formhosting\ Forminheritance \ Frames\GridPanel \ HTML\ ImageZoom \ IndexedDB \ MainMenu \ MessageDialogs \ Multiform \ PaintBox \ Pictures \ PushNotifications \ Regular Expressions \ ResponsiveGrid \ ResponsiveGridPanel \ RichEditor \ Simple \ ableControl \ Themes \ Treeview \ Accordion\Upload \ WebCrypto

**Lazarus206KbmMemtable_ChangeTool**
**https://www.blaisepascalmagazine.eu/wp-content/uploads/2019/12/KbmMemtable_ChangeTool.zip**

```
unit uActivityIndicator;

interface

uses
 Winapi.Windows,
 Winapi.Messages,
 System.SysUtils,
 System.Variants,
 System.Classes,
 System.ImageList,
 Vcl.Graphics,
 Vcl.Controls,
 Vcl.Forms,
 Vcl.Dialogs,
 Vcl.WinXCtrls,
 Vcl.StdCtrls,
 Vcl.ImgList,
 Vcl.ComCtrls,
 Vcl.ExtCtrls, Vcl.Imaging.pngimage;

type
 TActivityIndicatorForm = class(TForm)
  trkFrameDelay: TTrackBar;
  lblFrameDelay: TLabel;
  grpIndicatorType: TRadioGroup;
  grpIndicatorSize: TRadioGroup;
  grpIndicatorColor: TRadioGroup;
  cbxVclStyles: TComboBox;
  lblVclStyle: TLabel;
  AI: TActivityIndicator;
  cbxFormColor: TColorBox;
  lblFormColor: TLabel;
  Image1: TImage;
  Label1: TLabel;
  Label2: TLabel;
  OpenDialog1: TOpenDialog;
  ActivityIndicator1: TActivityIndicator;
  Label3: TLabel;
  procedure FormCreate(Sender: TObject);
  procedure trkFrameDelayChange(Sender: TObject);
  procedure grpIndicatorTypeClick(Sender: TObject);
  procedure grpIndicatorSizeClick(Sender: TObject);
  procedure grpIndicatorColorClick(Sender: TObject);
  procedure cbxVclStylesChange(Sender: TObject);
  procedure cbxFormColorChange(Sender: TObject);
  procedure Label1Click(Sender: TObject);
  procedure Label2Click(Sender: TObject);
  procedure Label3Click(Sender: TObject);
 private
 public
 end;

var
 ActivityIndicatorForm: TActivityIndicatorForm;

implementation

{$R *.dfm}

uses
 Vcl.Themes;
```

```
procedure TActivityIndicatorForm.FormCreate(Sender: TObject);
var StyleName: string;
begin
 Ai.Visible     := False;
 Image1.Visible := False;
 OpenDialog1.InitialDir := (ExtractFilePath(application.ExeName) + 'Images' );

 for StyleName in TStyleManager.StyleNames do  cbxVclStyles.Items.Add(StyleName);

 cbxVclStyles.ItemIndex := cbxVclStyles.Items.IndexOf(TStyleManager.ActiveStyle.Name);
end;

procedure TActivityIndicatorForm.cbxFormColorChange(Sender: TObject);
begin
 Color := cbxFormColor.Selected;
end;

procedure TActivityIndicatorForm.cbxVclStylesChange(Sender: TObject);
begin
 TStyleManager.SetStyle(cbxVclStyles.Text);
 lblFormColor.Enabled := StyleServices.IsSystemStyle;
 cbxFormColor.Enabled := StyleServices.IsSystemStyle;
end;

procedure TActivityIndicatorForm.grpIndicatorColorClick(Sender: TObject);
begin
 AI.IndicatorColor := TActivityIndicatorColor(grpIndicatorColor.ItemIndex);
end;

procedure TActivityIndicatorForm.grpIndicatorSizeClick(Sender: TObject);
begin
 AI.IndicatorSize := TActivityIndicatorSize(grpIndicatorSize.ItemIndex);
end;

procedure TActivityIndicatorForm.grpIndicatorTypeClick(Sender: TObject);
begin
 AI.IndicatorType := TActivityIndicatorType(grpIndicatorType.ItemIndex);
end;

procedure TActivityIndicatorForm.Label1Click(Sender: TObject);
begin
 Image1.Visible := Not Image1.Visible;
 AI.Visible     := Not Ai.Visible ;
 AI.Animate     := Not AI.Animate;
end;

procedure TActivityIndicatorForm.Label2Click(Sender: TObject);
begin
 if OpenDialog1.Execute then
   Image1.Picture.LoadFromFile(OpenDialog1.FileName);
end;

procedure TActivityIndicatorForm.Label3Click(Sender: TObject);
begin
 ActivityIndicator1.Animate    := Not ActivityIndicator1.Animate;
end;

procedure TActivityIndicatorForm.trkFrameDelayChange(Sender: TObject);
begin
 AI.FrameDelay := trkFrameDelay.Position * 10;
 lblFrameDelay.Caption := 'Frame Delay (' + IntToStr(AI.FrameDelay) + ')';
end;

end.
```

# The Lazarus Factory

## WHAT DOES THE LAZARUS FACTORY?

1. SUPPORTING COMPANY'S AND USERS ON A COMMERCIAL BASIS
2. DEVELOPING AND PROVIDING COURSES AND INSTRUCTION METHODS
3. CONTRIBUTING FINANCIALLY AND TECHNICALLY TO FREEPASCAL AND LAZARUS

- LAZARUS COMPILES FOR THE WEB USING PAS2JS
- LAZARUS HAS EMBEDDED WEBASSEMBLY
- LAZARUS NOW HAS GENERICS: FOR FPC, LAZARUS AND PAS2JS
- LAZARUS CAN COMPILE TO MULTIPLE PLATFORMS: WINDOWS / LINUX / MAC RASBERRY AND MORE
- LAZARUS USES NATIVE COMPONENTS
- LAZARUS CAN BE USED TO COMPILE DELPHI CODE. IF NOT LAZARUS FACTORY CAN HELP
- LAZARUS HAS A HANDBOOK OF ABOUT 1000 PAGES.

The Lazarus Factory

Free Pascal
Lazarus
Project
Write Once
Compile Anywhere
2.0.6

THE FREE COMPILER AND IDE

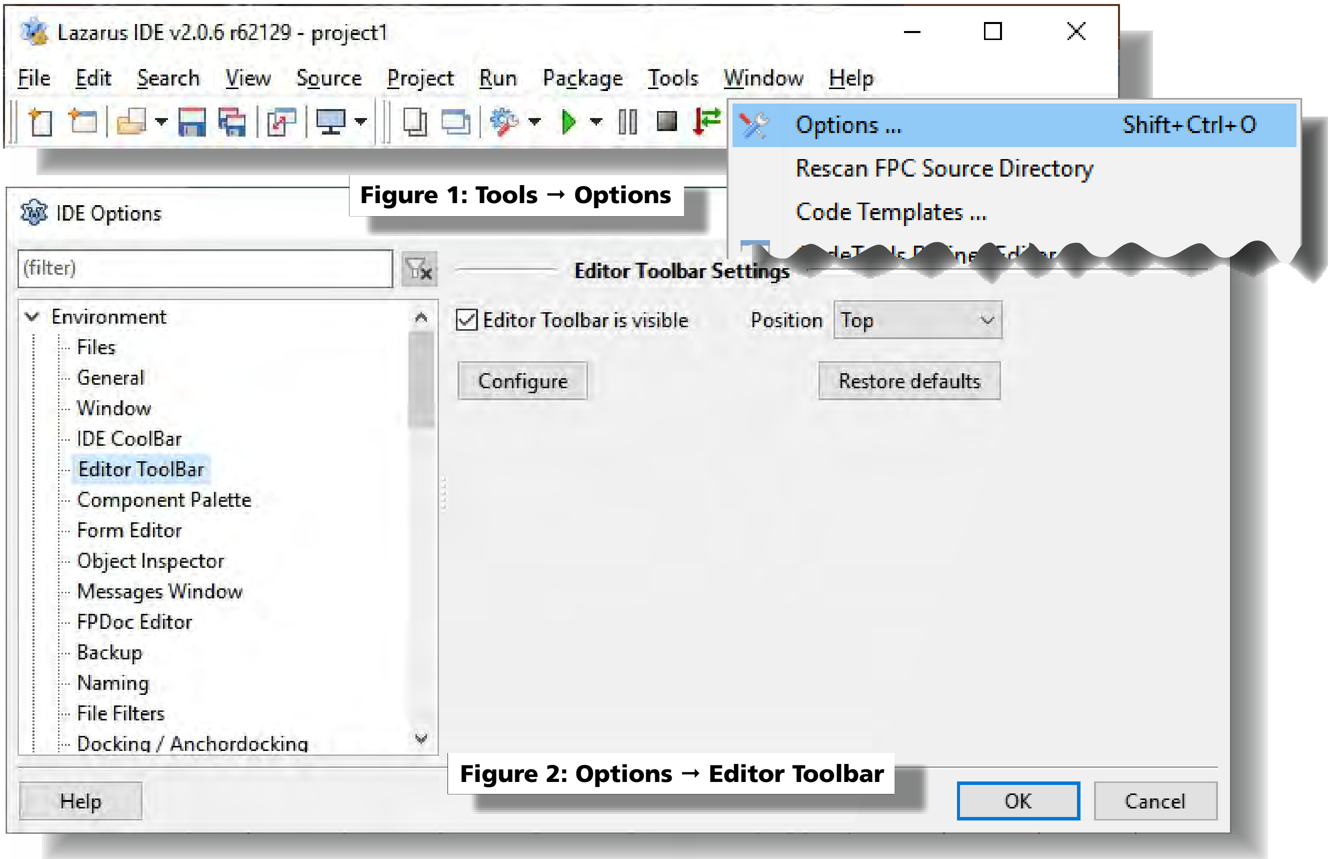
The Lazarus Factory

# LAZARUS: MOVING THE COMPONENT PALETTE PAGE 1/7

BY DETLEF OVERBEEK

SHORT HOWTO

MOVING / COOLBAR /



☑ Editor Toolbar is visible

Position: Top

**Figure 1: Tools → Options**

**Figure 2: Options → Editor Toolbar**

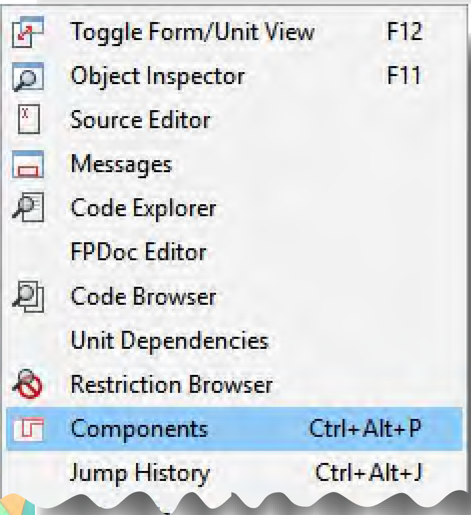


**Figure 3: Menu „View" → Components check the item Components**

## INTRODUCTION:

I want to show here how you can alter the place where the Component Palette is visible. Moving the **Component Palette** is actually very easy, but as in Delphi not located at the obvious place - where you would expect it. There are two possible candidates: the Editor Toolbar or the Component Palette. Neither one of them does what we want.

So I went for the menu **View:** here you find it: **Components** ( see left Figure 3). The form opens (below Figure 4) and that is what I wanted. But now I still need to get rid of the standard installed Components at the topbar.

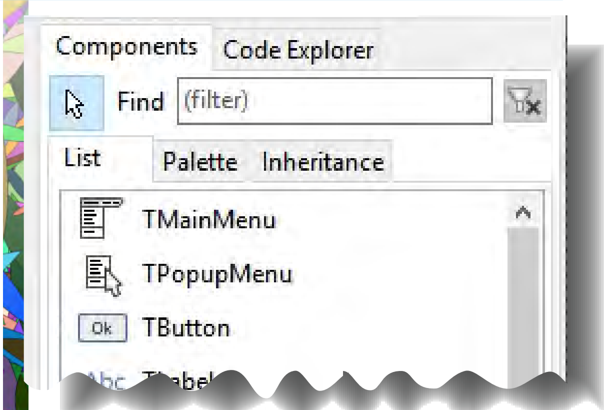


**Figure 4: The new Components window**

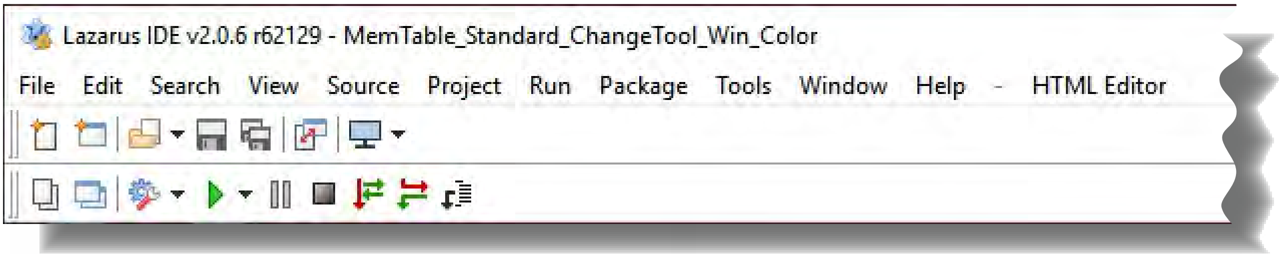

Lazarus IDE v2.0.6 r62129 - MemTable_Standard_ChangeTool_Win_Color
File Edit Search View Source Project Run Package Tools Window Help - HTML Editor

**Figure 5: The Coolbar standard devided over two lines**

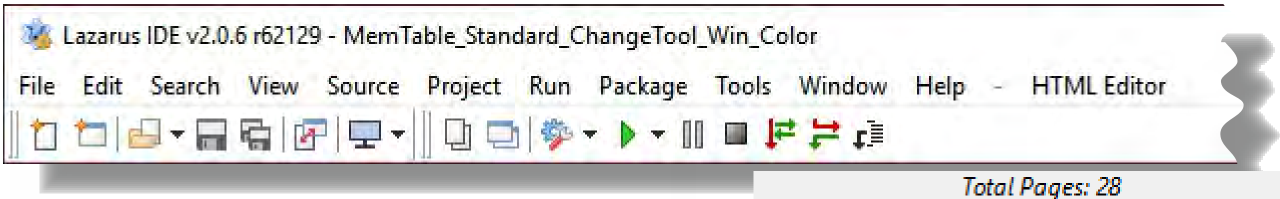

Lazarus IDE v2.0.6 r62129 - MemTable_Standard_ChangeTool_Win_Color
File Edit Search View Source Project Run Package Tools Window Help - HTML Editor

**Figure 6: The Coolbar can be set to one line**

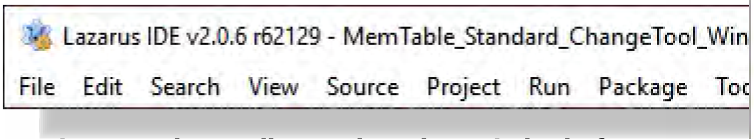

Lazarus IDE v2.0.6 r62129 - MemTable_Standard_ChangeTool_Win
File Edit Search View Source Project Run Package Too

**Figure 7: The Coolbar and can be switched of**

If you want to have no coolbar at all chose for set the Coolbar to invisible See the figure 10 below...

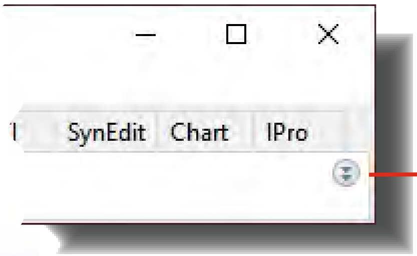
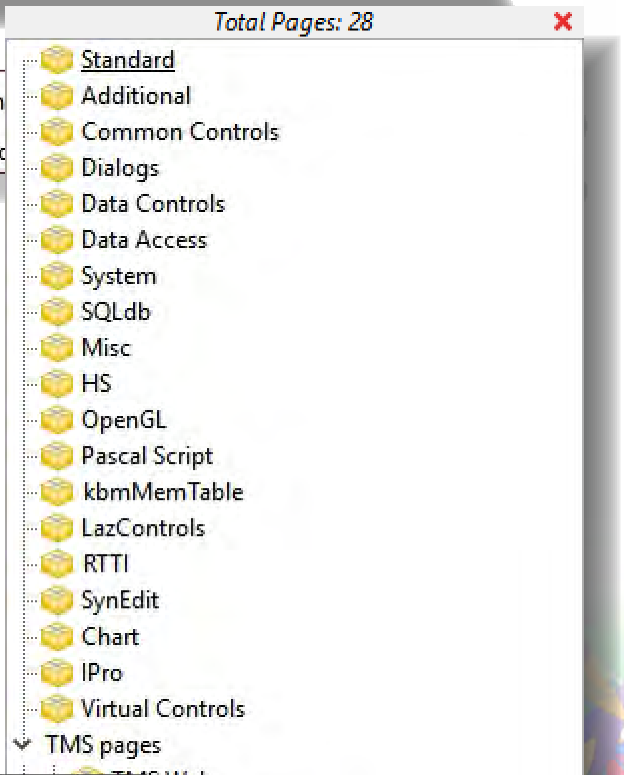


**Figure 8: If you click on the double arrow the left window will popup**

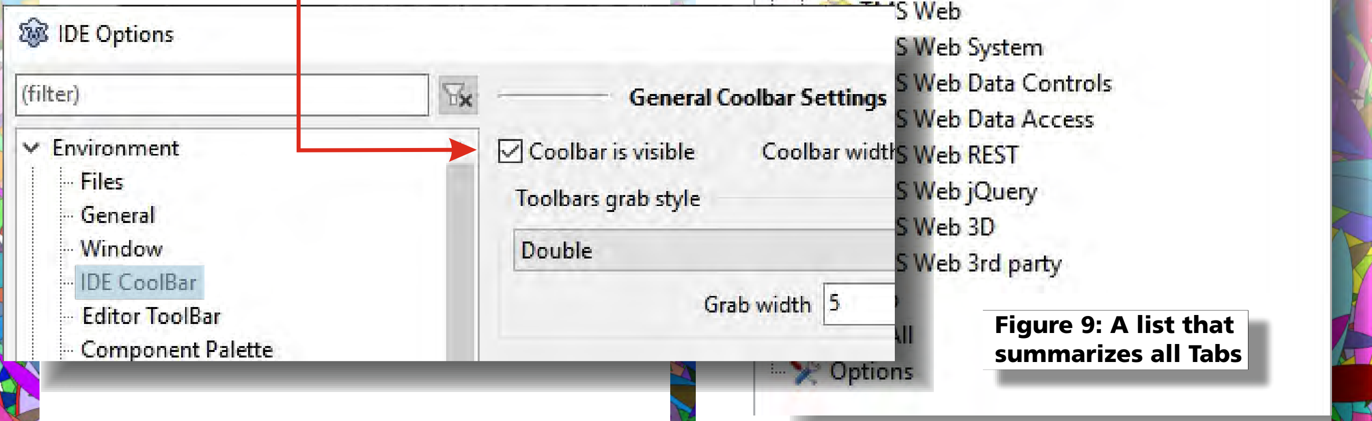


*Total Pages: 28*

- Standard
- Additional
- Common Controls
- Dialogs
- Data Controls
- Data Access
- System
- SQLdb
- Misc
- HS
- OpenGL
- Pascal Script
- kbmMemTable
- LazControls
- RTTI
- SynEdit
- Chart
- IPro
- Virtual Controls
- TMS pages
  - S Web
  - S Web System
  - S Web Data Controls
  - S Web Data Access
  - S Web REST
  - S Web jQuery
  - S Web 3D
  - S Web 3rd party
- ll
- Options

**Figure 9: A list that summarizes all Tabs**

**Figure 10: Uncheck the Coolbar**



IDE Options

(filter)

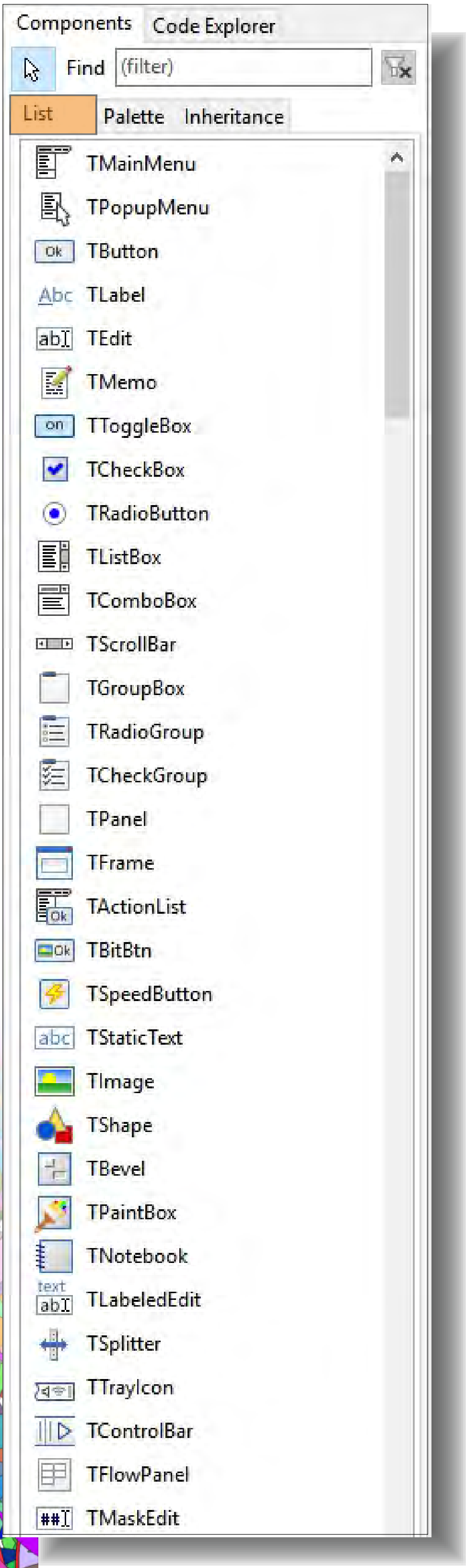
- Environment
  - Files
  - General
  - Window
  - IDE CoolBar
  - Editor ToolBar
  - Component Palette

General Coolbar Settings

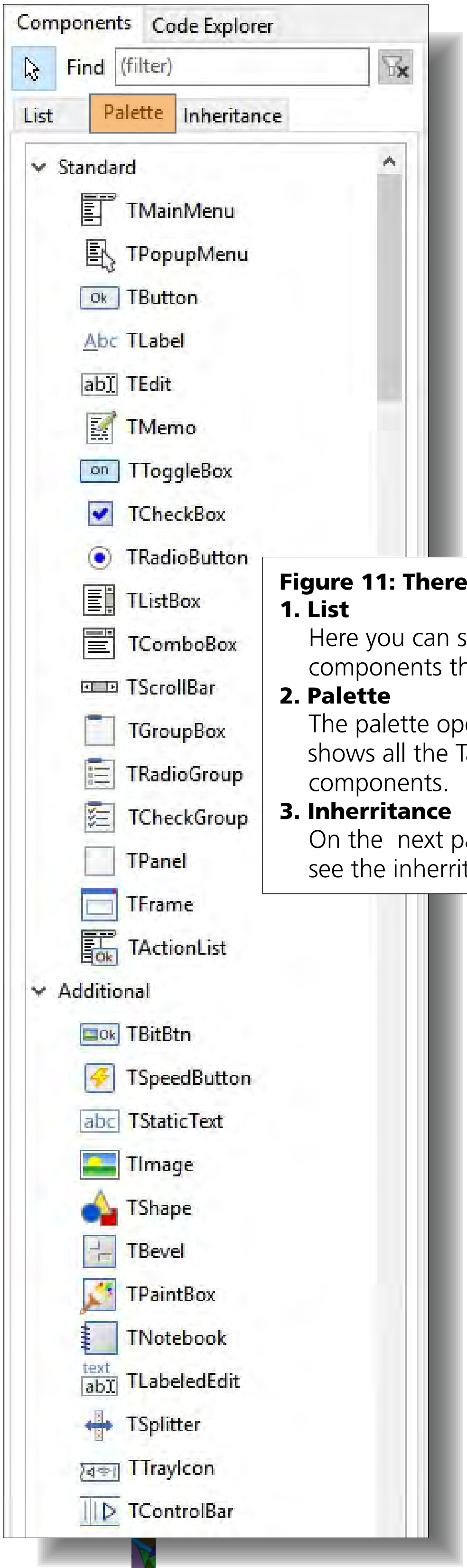
☑ Coolbar is visible    Coolbar widt

Toolbars grab style: Double

Grab width: 5



TMainMenu
TPopupMenu
TButton
TLabel
TEdit
TMemo
TToggleBox
TCheckBox
TRadioButton
TListBox
TComboBox
TScrollBar
TGroupBox
TRadioGroup
TCheckGroup
TPanel
TFrame
TActionList
TBitBtn
TSpeedButton
TStaticText
TImage
TShape
TBevel
TPaintBox
TNotebook
TLabeledEdit
TSplitter
TTrayIcon
TControlBar
TFlowPanel
TMaskEdit



Standard
TMainMenu
TPopupMenu
TButton
TLabel
TEdit
TMemo
TToggleBox
TCheckBox
TRadioButton
TListBox
TComboBox
TScrollBar
TGroupBox
TRadioGroup
TCheckGroup
TPanel
TFrame
TActionList

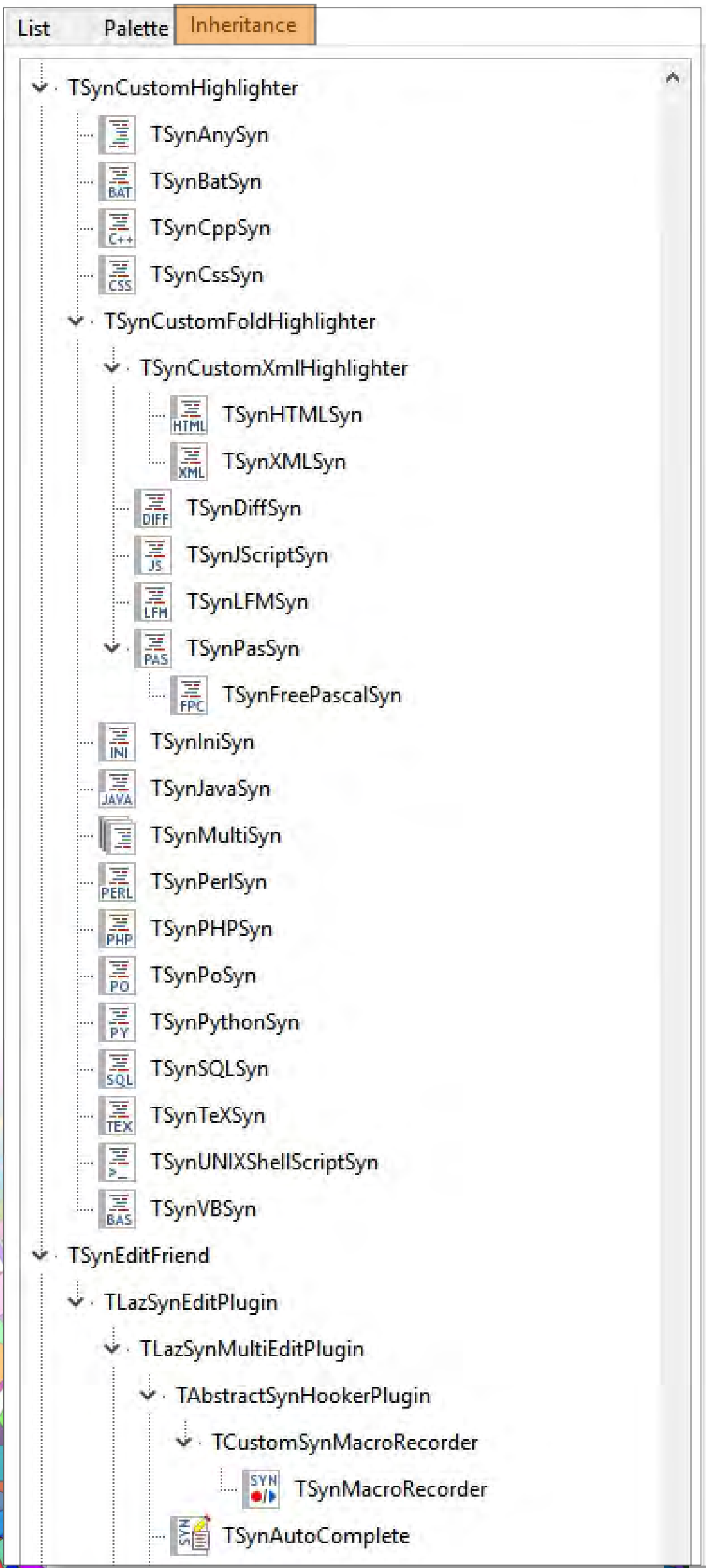
Additional
TBitBtn
TSpeedButton
TStaticText
TImage
TShape
TBevel
TPaintBox
TNotebook
TLabeledEdit
TSplitter
TTrayIcon
TControlBar

**Figure 11: There are three Tabs:**

**1. List**
Here you can search and find all the components that are available

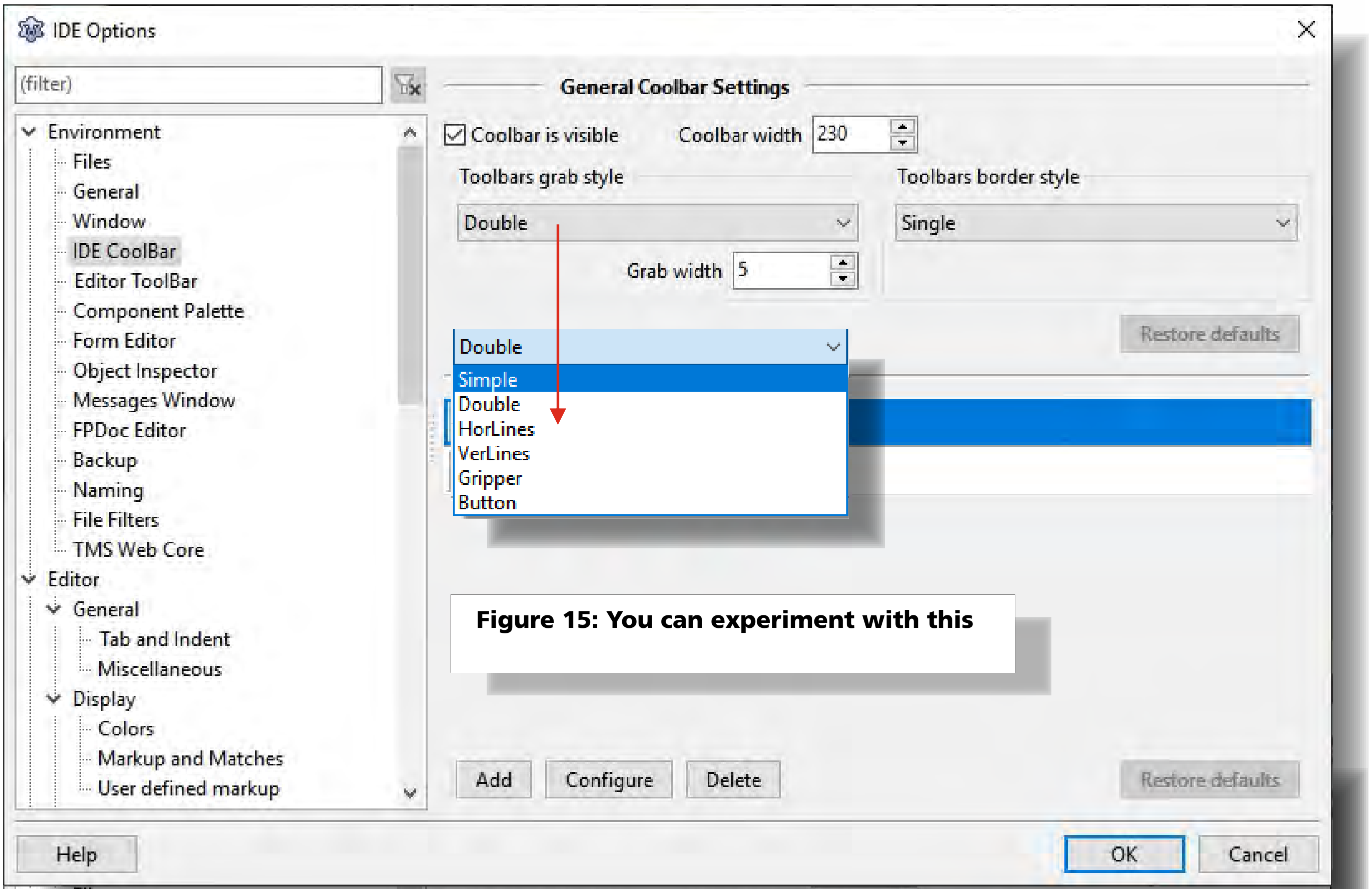
**2. Palette**
The palette opens the separated Tabs shows all the Tabs and and their components.

**3. Inherritance**
On the next page (figure 12) you can see the inherritance of the components.



```
TSynCustomHighlighter
    TSynAnySyn
    TSynBatSyn
    TSynCppSyn
    TSynCssSyn
    TSynCustomFoldHighlighter
        TSynCustomXmlHighlighter
            TSynHTMLSyn
            TSynXMLSyn
        TSynDiffSyn
        TSynJScriptSyn
        TSynLFMSyn
        TSynPasSyn
            TSynFreePascalSyn
    TSynIniSyn
    TSynJavaSyn
    TSynMultiSyn
    TSynPerlSyn
    TSynPHPSyn
    TSynPoSyn
    TSynPythonSyn
    TSynSQLSyn
    TSynTeXSyn
    TSynUNIXShellScriptSyn
    TSynVBSyn
TSynEditFriend
    TLazSynEditPlugin
        TLazSynMultiEditPlugin
            TAbstractSynHookerPlugin
                TCustomSynMacroRecorder
                    TSynMacroRecorder
            TSynAutoComplete
```

**Figure 12: An overview of the coponents together with their ancestry**



IDE Options

(filter)

- Environment
  - Files
  - General
  - Window
  - IDE CoolBar
  - Editor ToolBar
  - Component Palette
  - Form Editor
  - Object Inspector
  - Messages Window
  - FPDoc Editor
  - Backup
  - Naming
  - File Filters
  - TMS Web Core
- Editor
  - General
    - Tab and Indent
    - Miscellaneous
  - Display
    - Colors
    - Markup and Matches
    - User defined markup

General Coolbar Settings

☑ Coolbar is visible  Coolbar width 230

Toolbars grab style: Double

Toolbars border style: Single

Grab width 5

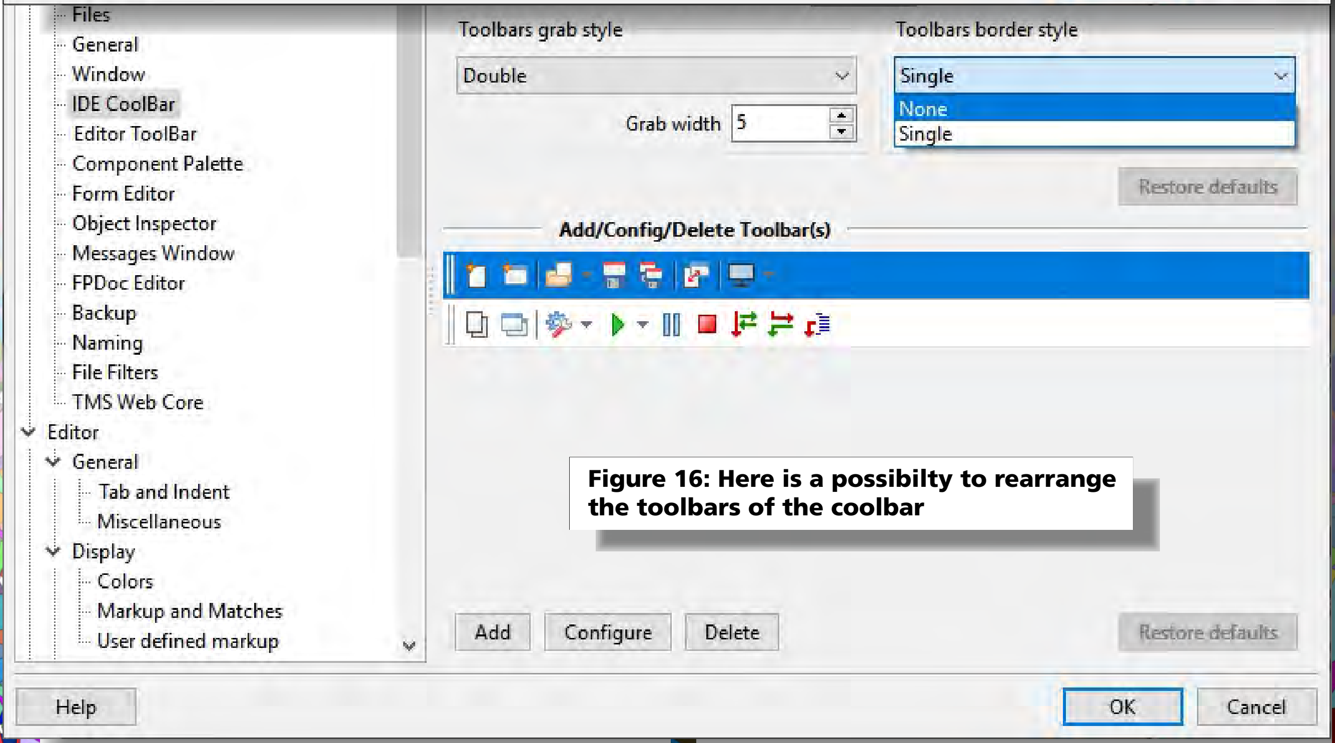
Double
- Simple
- Double
- HorLines
- VerLines
- Gripper
- Button

Restore defaults

**Figure 15: You can experiment with this**

Add Configure Delete Restore defaults

Help OK Cancel



- Files
- General
- Window
- IDE CoolBar
- Editor ToolBar
- Component Palette
- Form Editor
- Object Inspector
- Messages Window
- FPDoc Editor
- Backup
- Naming
- File Filters
- TMS Web Core
- Editor
  - General
    - Tab and Indent
    - Miscellaneous
  - Display
    - Colors
    - Markup and Matches
    - User defined markup

Toolbars grab style: Double

Toolbars border style: Single
- None
- Single

Grab width 5

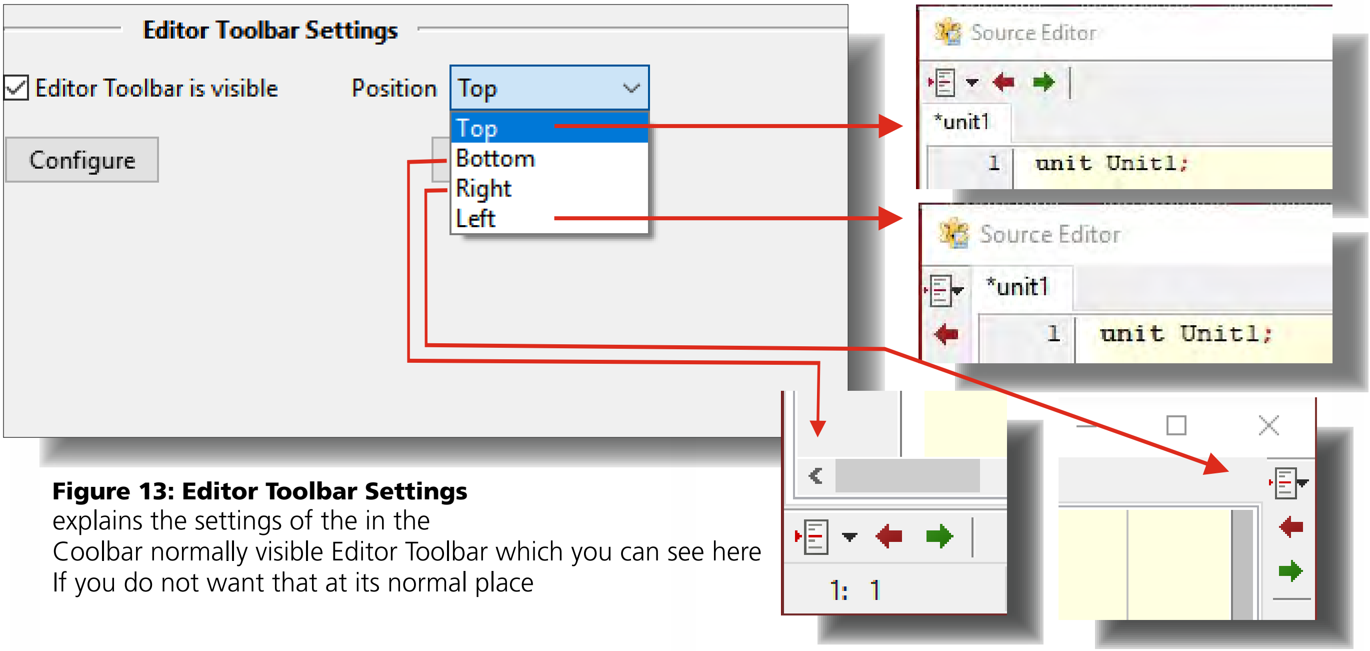
Restore defaults

Add/Config/Delete Toolbar(s)

**Figure 16: Here is a possibilty to rearrange the toolbars of the coolbar**

Add Configure Delete Restore defaults

Help OK Cancel

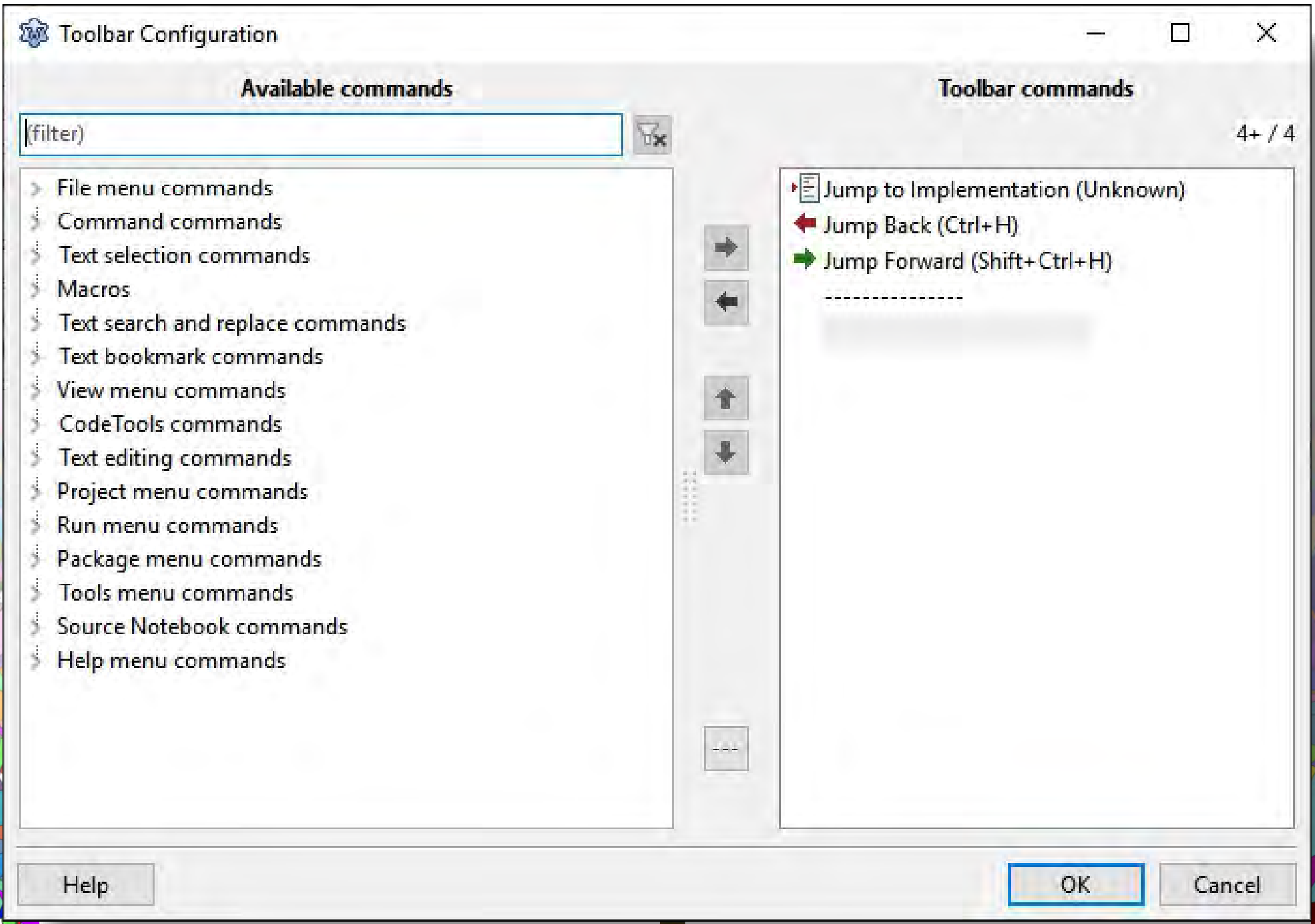


Editor Toolbar Settings

☑ Editor Toolbar is visible

Position: Top

**Figure 13: Editor Toolbar Settings**
explains the settings of the in the
Coolbar normally visible Editor Toolbar which you can see here
If you do not want that at its normal place



Toolbar Configuration

Available commands

(filter)

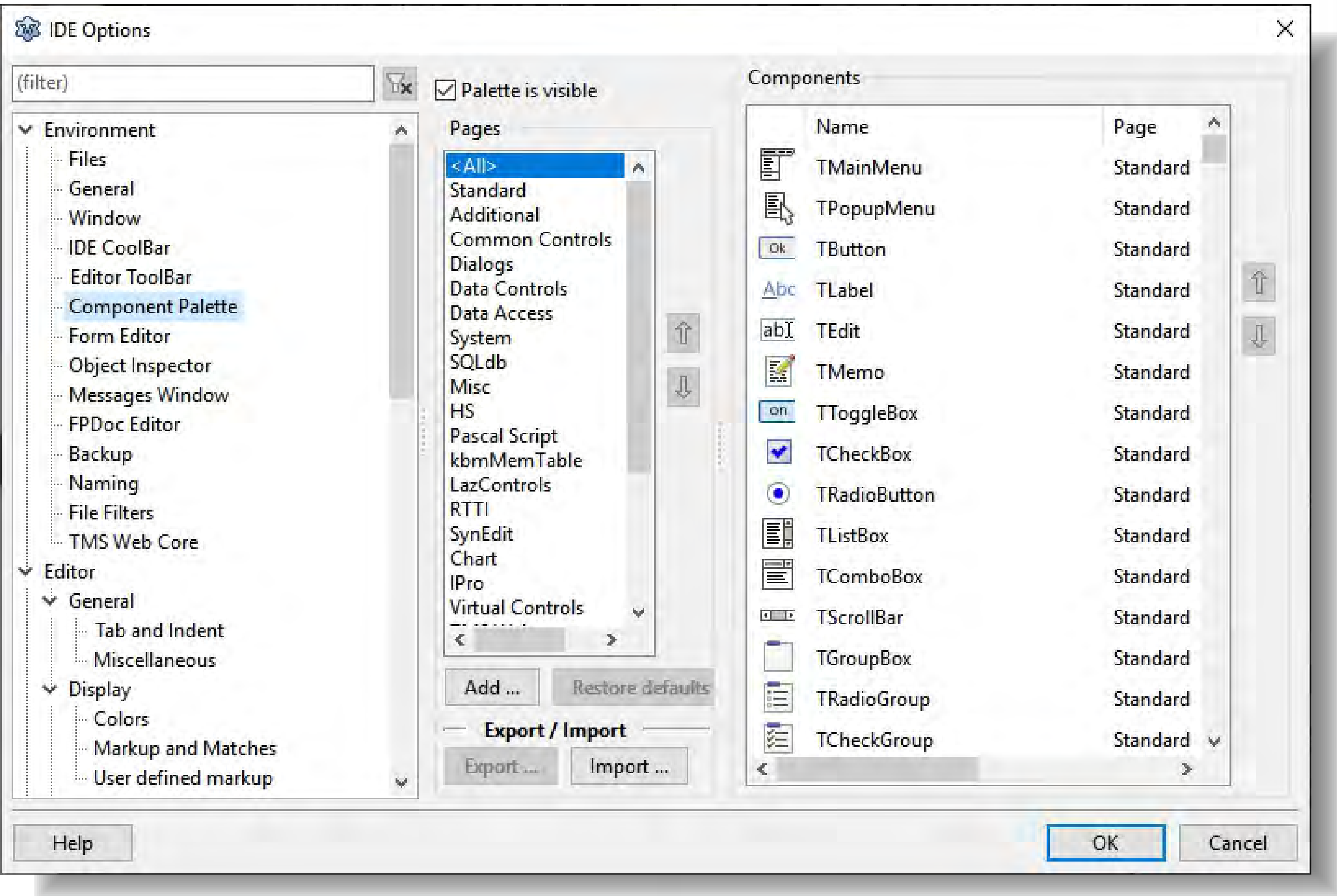
- File menu commands
- Command commands
- Text selection commands
- Macros
- Text search and replace commands
- Text bookmark commands
- View menu commands
- CodeTools commands
- Text editing commands
- Project menu commands
- Run menu commands
- Package menu commands
- Tools menu commands
- Source Notebook commands
- Help menu commands

Toolbar commands

4+ / 4

- Jump to Implementation (Unknown)
- Jump Back (Ctrl+H)
- Jump Forward (Shift+Ctrl+H)
- ---------------

**Figure 14: You can rearrange the components**



IDE Options

(filter)

- Environment
  - Files
  - General
  - Window
  - IDE CoolBar
  - Editor ToolBar
  - Component Palette
  - Form Editor
  - Object Inspector
  - Messages Window
  - FPDoc Editor
  - Backup
  - Naming
  - File Filters
  - TMS Web Core
- Editor
  - General
    - Tab and Indent
    - Miscellaneous
  - Display
    - Colors
    - Markup and Matches
    - User defined markup

☑ Palette is visible

Pages

<All>
Standard
Additional
Common Controls
Dialogs
Data Controls
Data Access
System
SQLdb
Misc
HS
Pascal Script
kbmMemTable
LazControls
RTTI
SynEdit
Chart
IPro
Virtual Controls

Add ... Restore defaults

Export / Import

Export ... Import ...

Components

| Name | Page |
|---|---|
| TMainMenu | Standard |
| TPopupMenu | Standard |
| TButton | Standard |
| TLabel | Standard |
| TEdit | Standard |
| TMemo | Standard |
| TToggleBox | Standard |
| TCheckBox | Standard |
| TRadioButton | Standard |
| TListBox | Standard |
| TComboBox | Standard |
| TScrollBar | Standard |
| TGroupBox | Standard |
| TRadioGroup | Standard |
| TCheckGroup | Standard |

Help OK Cancel

**Figure 17: The component palette is very much alike Delphi.**

barnsten Locatie: Landgoed Jachthuis Beukenrode in Doorn (bij Utrecht)

BOUW JE EIGEN VCL APPLICATIES VOOR WINDOWS 10

# Delphi VCL Essentials Training

We starten het jaar goed met een nieuwe driedaagse Delphi VCL Essentials training! Deze populaire training wordt verzorgd door trainer Danny Wind (Delphi MVP) en is inclusief zeer uitgebreid Nederlandstalig les- en oefenmateriaal.

Voor wie is deze training:

- Voor coders en ontwikkelaars, met enige programmeer ervaring in Delphi of een andere taal, die graag snel en goed met Delphi aan de slag willen.
- Voor Delphi ontwikkelaars die willen overstappen van een oudere versie naar de nieuwe 10.3 Rio versie en hun huidige kennis willen opfrissen.

De training begint met een overzicht van alles wat u met Delphi en de VCL kunt bouwen. Daarna gaat u de diepte in en leert u hoe u zelf VCL applicaties kunt bouwen en onderhouden. Hierbij komen ook zaken aan bod zoals werken met databases, vormgeven van applicaties, debuggen, touch en veel meer.

Wilt u ook alles uit uw Delphi omgeving halen en gebruik maken van de nieuwste functionaliteit? Schrijf dan nu in. De training wordt gehouden in Doorn (bij Utrecht) op 11 t/m 13 maart 2020.

***In verband met de kwaliteit van de training werken wij met kleine groepen en is het aantal plaatsen beperkt.***

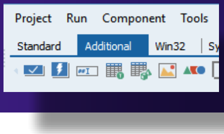
Heeft u vragen over de training of over Delphi software? Neem dan direct contact met ons op via 023 542 22 27 of per email operations@barnsten.com

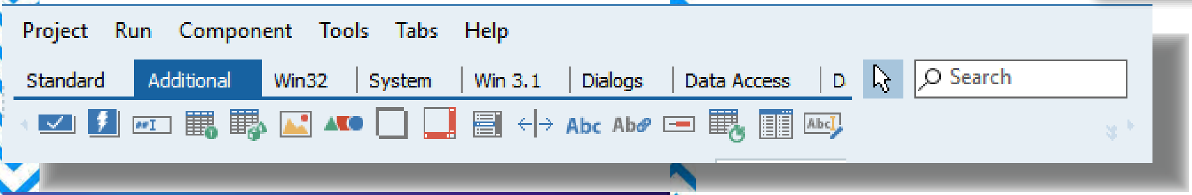
https://www.barnsten.com/nl/product/delphi-vcl-essentials-training/

# DELPHI: THE COMPONENT PALETTE REFURBISHED PAGE 1/2

BY DETLEF OVERBEEK

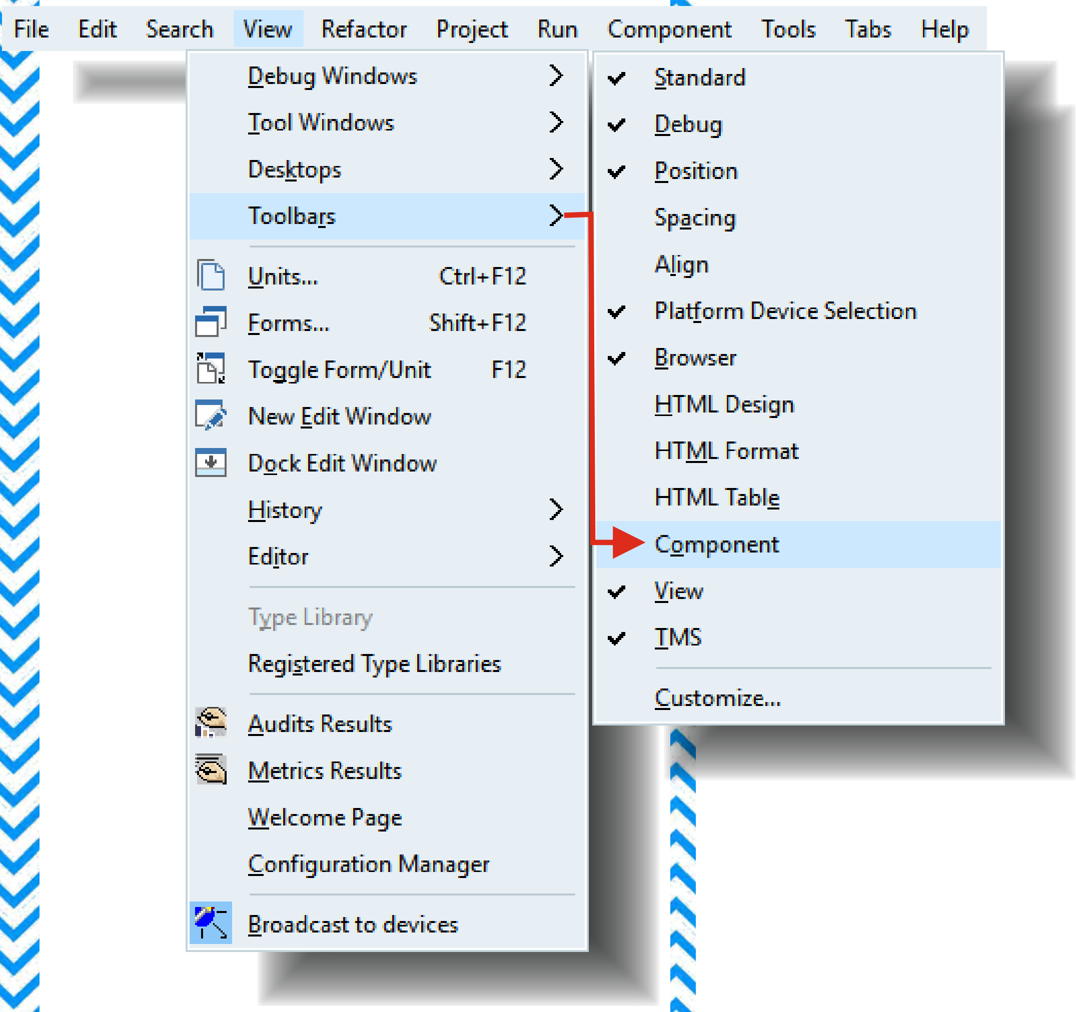
SHORT HOWTO

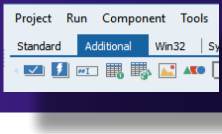


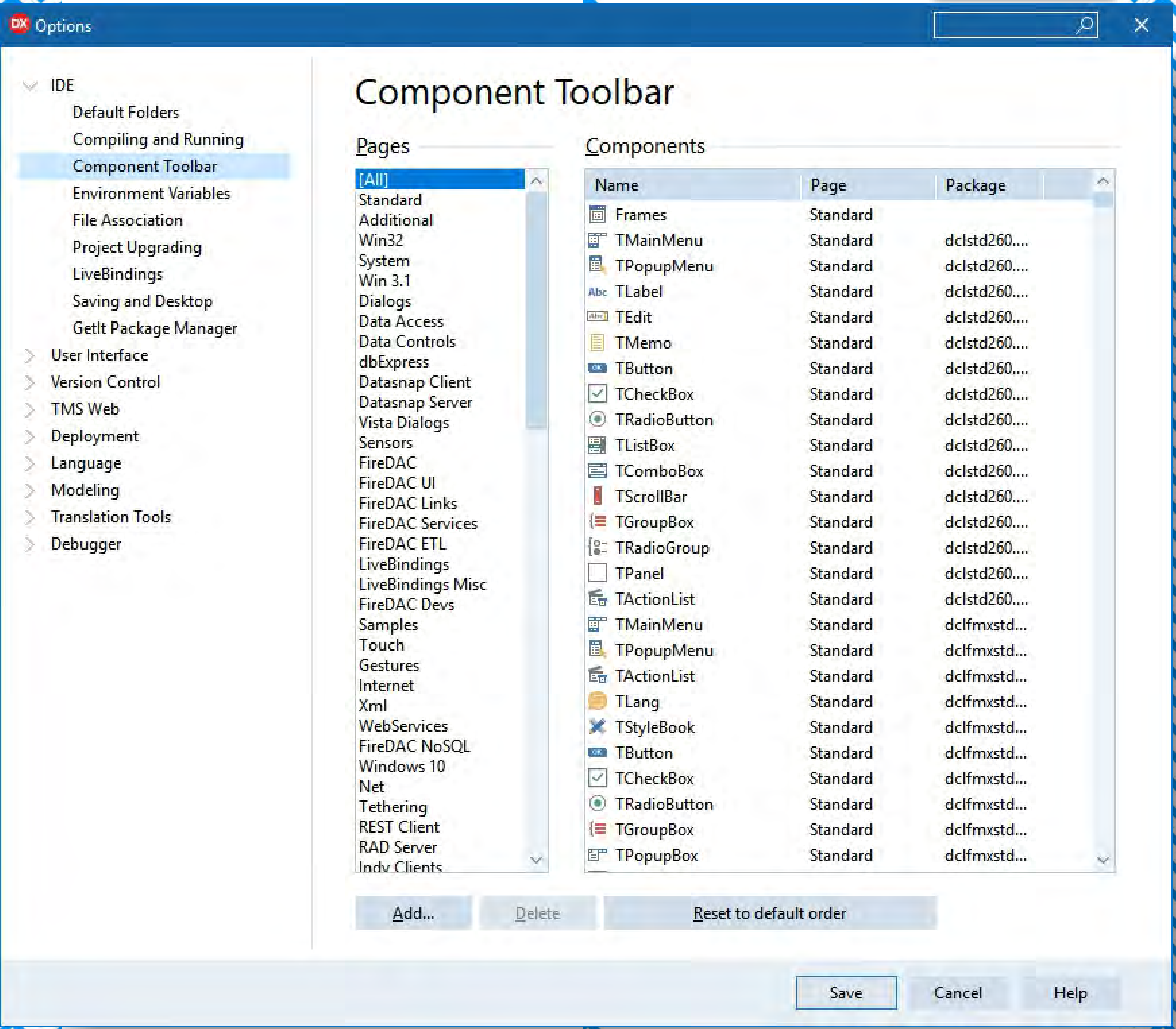

Project Run Component Tools Tabs Help
Standard Additional Win32 System Win 3.1 Dialogs Data Access D
Search
Abc Abc

In case you like to have your component palette available a the top, delphi made it quite easy to do. But.
As always you need to know where to find it.
It took me quite some time to track down:

Go to the menu: **View** and then → **Toolbars** → **Component.** To my surprise it immeddiatel apppears and is reday for use.
I would have expected it under **Tools** → **Component Toolbar.** (See next page)
Nothing there that you can change.







IDE
Default Folders
Compiling and Running
Component Toolbar
Environment Variables
File Association
Project Upgrading
LiveBindings
Saving and Desktop
GetIt Package Manager
User Interface
Version Control
TMS Web
Deployment
Language
Modeling
Translation Tools
Debugger

## Component Toolbar

Pages

[All]
Standard
Additional
Win32
System
Win 3.1
Dialogs
Data Access
Data Controls
dbExpress
Datasnap Client
Datasnap Server
Vista Dialogs
Sensors
FireDAC
FireDAC UI
FireDAC Links
FireDAC Services
FireDAC ETL
LiveBindings
LiveBindings Misc
FireDAC Devs
Samples
Touch
Gestures
Internet
Xml
WebServices
FireDAC NoSQL
Windows 10
Net
Tethering
REST Client
RAD Server
Indy Clients

Components

| Name | Page | Package |
|---|---|---|
| Frames | Standard | |
| TMainMenu | Standard | dclstd260.... |
| TPopupMenu | Standard | dclstd260.... |
| TLabel | Standard | dclstd260.... |
| TEdit | Standard | dclstd260.... |
| TMemo | Standard | dclstd260.... |
| TButton | Standard | dclstd260.... |
| TCheckBox | Standard | dclstd260.... |
| TRadioButton | Standard | dclstd260.... |
| TListBox | Standard | dclstd260.... |
| TComboBox | Standard | dclstd260.... |
| TScrollBar | Standard | dclstd260.... |
| TGroupBox | Standard | dclstd260.... |
| TRadioGroup | Standard | dclstd260.... |
| TPanel | Standard | dclstd260.... |
| TActionList | Standard | dclstd260.... |
| TMainMenu | Standard | dclfmxstd... |
| TPopupMenu | Standard | dclfmxstd... |
| TActionList | Standard | dclfmxstd... |
| TLang | Standard | dclfmxstd... |
| TStyleBook | Standard | dclfmxstd... |
| TButton | Standard | dclfmxstd... |
| TCheckBox | Standard | dclfmxstd... |
| TRadioButton | Standard | dclfmxstd... |
| TGroupBox | Standard | dclfmxstd... |
| TPopupBox | Standard | dclfmxstd... |

You can Add Page, reorganize and reset the order you have created. No more options are avilable. Searching in the environment offers no other options.

# COMPASSIST - THE NOTIFICATION CENTER COMPONENT PAGE 1/3

starter expert

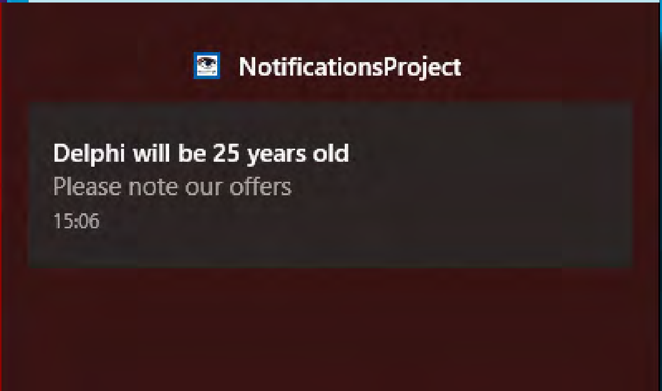
DX

COMPASSIST

In **Delphi Rio 10.3.3** there are some great items you should know. They will be shown here as projects so we can see how the component can be handled and what they are meant for. This is going to be an ongoing series and we will try to illuminate components that we think are extra interesting: **CompAssist**(Component Assistent)

This sample demonstrates how to create, present, and cancel notifications on Windows 8 or later.

This project uses a **`TNotificationCenter`** component, and was originatad by Embarcadero This is only available starting windows 8 >.



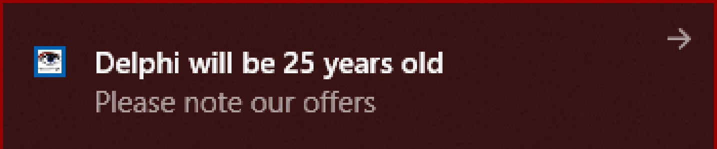
NotificationsProject

Delphi will be 25 years old
Please note our offers
15:06

**Figure1: The message appeard in the top right side**

## HOW TO USE THE APP

Press F9 or choose Run > Run.
Click the different buttons to create and delete notifications:
Click Show to present a notification.
Click Show Another to present a second notification.
Click Cancel to dismiss the notification presented after clicking Show.
Click Cancel Another to dismiss the notification presented after clicking Show Another.
Click Cancel All to cancel all the notifications.
Handle the notifications to show a received message in the TMemo.



Delphi will be 25 years old
Please note our offers

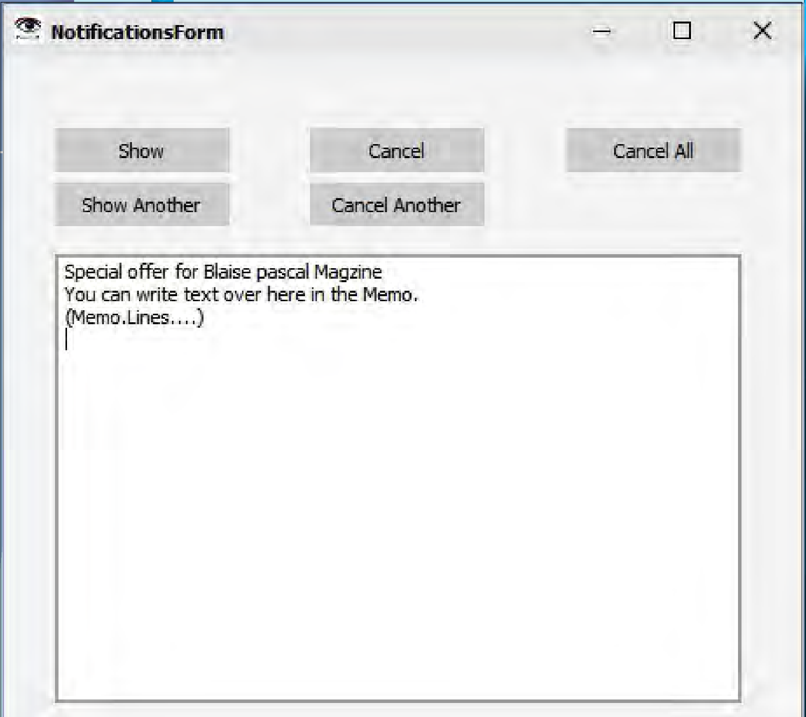
**Figure2: The message appeard in the right low corner**

## IMPLEMENTATION

The sample uses the **`TNotificationCenter`** component and handle notifications.
When a notification is presented in Windows, it appears as a banner in the right up/bottom side of the window. (See figure 1 and 2)

If the notification is not handled by the user, it goes to the Action Center (Only for Windows 10).

The sample uses the buttons Cancel, Cancel Another, and Cancel All to remove notifications from the Action Center.
When the notification is handled, that is when a user clicks it, the OnReceiveLocalNotification event triggers and a line is added to the TMemo.



```
Special offer for Blaise pascal Magzine
You can write text over here in the Memo.
(Memo.Lines....)
```

**Figure3: The project shows how to handle**

```
unit Notifications;

interface

uses
 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, System.Notification, Vcl.StdCtrls;

type
 TNotificationsForm = class(TForm)
  NotificationCenter1: TNotificationCenter;
  mmLog: TMemo;
  btnShow: TButton;
  btnCancel: TButton;
  btnCancelAll: TButton;
  btnShowAnother: TButton;
  btnCancelAnother: TButton;
  procedure NotificationCenter1ReceiveLocalNotification(Sender: TObject; ANotification: TNotification);
  procedure btnShowClick(Sender: TObject);
  procedure btnCancelClick(Sender: TObject);
  procedure btnCancelAllClick(Sender: TObject);
  procedure btnShowAnotherClick(Sender: TObject);
  procedure btnCancelAnotherClick(Sender: TObject);
  procedure FormShow(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var NotificationsForm: TNotificationsForm;

implementation

{$R *.dfm}

procedure TNotificationsForm.btnCancelAllClick(Sender: TObject);
begin
 NotificationCenter1.CancelAll;
end;

procedure TNotificationsForm.btnCancelAnotherClick(Sender: TObject);
begin
 NotificationCenter1.CancelNotification('Windows10Notification2');
end;

procedure TNotificationsForm.btnCancelClick(Sender: TObject);
begin
 NotificationCenter1.CancelNotification('Windows10Notification');
end;

procedure TNotificationsForm.btnShowAnotherClick(Sender: TObject);
var
 MyNotification: TNotification;
begin
 MyNotification := NotificationCenter1.CreateNotification;
 try
  MyNotification.Name := 'Windows10Notification2';
  MyNotification.Title := 'Windows 10 Notification #2';
  MyNotification.AlertBody := 'RAD Studio 10 Seattle';

  NotificationCenter1.PresentNotification(MyNotification);
 finally
  MyNotification.Free;
 end;
end;
```

```
procedure TNotificationsForm.btnShowClick(Sender: TObject);
var
 MyNotification: TNotification;
begin
 MyNotification := NotificationCenter1.CreateNotification;
 try
  MyNotification.Name := 'Windows10Notification';
  MyNotification.Title := 'Windows 10 Notification #1';
  MyNotification.AlertBody := 'RAD Studio 10 Seattle';

  NotificationCenter1.PresentNotification(MyNotification);
 finally
  MyNotification.Free;
 end;
end;

procedure TNotificationsForm.FormShow(Sender: TObject);
begin
 OnShow := nil;
{$IFDEF MSWINDOWS}
 if not TOSVersion.Check(6, 2) then // Windows 8
 begin
  ShowMessage('This demo is designed to show Notification feature in Windows 8 or higher. Bye.');
  Application.Terminate;
 end;
{$ENDIF MSWINDOWS}
end;

procedure TNotificationsForm.NotificationCenter1ReceiveLocalNotification(Sender: TObject;
 ANotification: TNotification);
begin
 mmLog.Lines.Add('Notification received: ' + ANotification.Name);
end;

end.
```

By Onderwijsgek - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=19789305

# KBMMW PROFESSIONAL AND ENTERPRISE EDITION V. 5.10.20 RELEASED!

- RAD Studio XE2 to 10.3 Rio supported
- Win32, Win64, Linux64, Android, IOS 32, IOS 64 and OSX client and server support
- Native high performance 100% developer defined application server
- Full support for centralized and distributed load balancing and failover
- Advanced ORM/OPF support including support of existing databases
- Advanced logging support
- Advanced configuration framework
- Advanced scheduling support for easy access to multithread programming
- Advanced smart service and clients for very easy publication of functionality
- High quality random functions.
- High quality pronouncable password generators.
- High performance LZ4 and Jpeg compression
- Complete object notation framework including full support for YAML, BSON, Messagepack, JSON and XML
- Advanced object and value marshalling to and from YAML, BSON, Messagepack, JSON and XML
- High performance native TCP transport support
- High performance HTTPSys transport for Windows.
- CORS support in REST/HTML services.
- Native PHP, Java, OCX, ANSI C, C#, Apache Flex client support!

**kbmMemTable is the fastest and most feature rich in memory table for Embarcadero products.**

- **Easily supports large datasets with millions of records**
- **Easy data streaming support**
- **Optional to use native SQL engine**
- **Supports nested transactions and undo**
- **Native and fast build in M/D, aggregation/grouping, range selection features**
- **Advanced indexing features for extreme performance**

♦ **NEW! SmartBind now fully supports VCL, FMX, including image/graphics and TListView**

♦ **NEW! SmartBind data generators and data proxies for easy separation of data sharing concerns in modular applications**

♦ **NEW! SmartEvent for easy separation of event and execution workflow based concerns for the ultimate in modular application design**

♦ **NEW! Native highly scalable TCP server transport now also supports REST**

♦ **Significant improvements and fixes in many areas including**
◊ **RTTI**
◊ **Scheduler**
◊ **LINQ**
◊ **Object Notation**
◊ **ORM**

- High speed, unified database access (35+ supported database APIs) with connection pooling, metadata and data caching on all tiers
- Multi head access to the application server, via REST/AJAX, native binary, Publish/Subscribe, SOAP, XML, RTMP from web browsers, embedded devices, linked application servers, PCs, mobile devices, Java systems and many more clients
- Complete support for hosting FastCGI based applications (PHP/Ruby/Perl/Python typically)
- Native complete AMQP 0.91 support (Advanced Message Queuing Protocol)
- Complete end 2 end secure brandable Remote Desktop with near realtime HD video, 8 monitor support, texture detection, compression and clipboard sharing.
- Bundling kbmMemTable Professional which is the fastest and most feature rich in memory table for Embarcadero products.

COMPONENTS 4 DEVELOPERS

DX

**EESB, SOA,MoM, EAI TOOLS FOR INTELLIGENT SOLUTIONS. kbmMW IS THE PREMIERE N-TIER PRODUCT FOR DELPHI / C++BUILDER**